Learning from Simulated World - Surrogates Construction with Deep Neural Network

Zong-De Jian[1], Hung-Jui Chang[1,2], Tsan-sheng Hsu[1] and Da-Wei Wang[1]
[1]*Institute of Information Science, Academia Sinica, Taipei, Taiwan*
[2]*Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan*

Keywords: Deep Learning, Surrogate, Disease Simulator.

Abstract: The deep learning approach has been applied to many domains with success. We use deep learning to construct the surrogate function to speed up simulation based optimization in epidemiology. The simulator is an agent-based stochastic model for influenza and the optimization problem is to find vaccination strategy to minimize the number of infected cases. The optimizer is a genetic algorithm and the fitness function is the simulation program. The simulation is the bottleneck of the optimization process. An attempt to use the surrogate function with table lookup and interpolation was reported before. The preliminary results show that the surrogate constructed by deep learning approach outperforms the interpolation based one, as long as *similar* cases of the testing set have been available in the training set. The average of the absolute value of relative error is less than 0.7 percent, which is quite close to the intrinsic limitation of the stochastic variation of the simulation software 0.2 percent, and the rank coefficients are all above 0.99 for cases we studied. The vaccination strategy recommended is still to vaccine the school age children first which is consistent with the previous studies. The preliminary results are encouraging and it should be a worthy effort to use machine learning approach to explore the vast parameter space of simulation models in epidemiology.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

Simulation models are built so that we can experiment with them to gain insight about subjects investigated. With the current success of machine learning, especially deep learning, it is worth exploring about using machine learning technique to learn from the simulation models. We explore the possibility by using the deep neural network to construct a surrogate function as the cost function instead of running real simulation when applying the genetic algorithm to search for effective vaccination strategy in the domain of public health.

Agent-based stochastic simulations have been applied widely for the study of infectious diseases (Germann et al., 2006)(Chao et al., 2010). Comparing to the mathematical models, the flexibility to easily incorporate detailed disease control strategies into simulation model is one of its strength. However, it still needs a significant amount of computing resources. Epidemiologists usually have to carefully craft the scenarios to demonstrate their points. Vaccination can effectively mitigate the impact of the pandemic flu with an appropriate vaccinating strategy which might depend on an amount of available vaccines. Instead of comparing a few strategies selected by domain experts, we formulate it as an optimization problem and employ genetic algorithms to search for the best vaccination priority. The search space can potentially contain many dimensions, for example, the house-hold structure is one of the important dimensions (Chang et al., 2015). In this paper, we focus on the dimension of vaccination allocation.

The supply of the vaccine is usually limited, the disease control agency has to decide the amount of vaccine allocated to various groups. Without a doubt, the health care professionals should get the highest priority. Then the agency determined how much to distribute to different age groups. How to determine the number of doses to each age group is an important issue. The objective can also be complicated, for example, one can search for a strategy to minimize economic impact, or to minimize the total number of infected cases. In this paper, the goal is to minimize the total number of infected cases.

For a given scenario, that is the setting of our simulation module, the gene encodes the vaccine distribution among age groups and the fitness function is the

Jian, Z-D., Chang, H-J., Hsu, T-s. and Wang, D-W.
Learning from Simulated World - Surrogates Construction with Deep Neural Network.
DOI: 10.5220/0006418100830092
In *Proceedings of the 7th International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2017)*, pages 83-92
ISBN: 978-989-758-265-3
Copyright © 2017 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

total number of infected cases. The fitness evaluation is done by running the simulation module.

Each simulation run takes about 3 minutes, thus the fitness evaluation becomes the bottleneck of the optimization process. Using a faster approximation in place of the true fitness function, in our case the simulation program, is called surrogated-assisted evolutionary computation (Jin, 2011). The idea was first suggested in the mid-1980s (J.J. Grefenstette, 1985). A surrogate function combines table lookups and linear interpolation was suggested in (Jian et al., 2016). In this study, we use the deep neural network to construct the surrogate function and compare it with the previous interpolation based one.

The accuracy is measured by the relative error, that is the difference between the output of surrogate function and the simulation divided by the output of the simulation. The average of the absolute value of the relative errors of the surrogate functions constructed by the deep neural network approach range from 0.18 percent to 0.68 percent for different training sets and testing sets. When the training sets have cases similar to the testing sets, the surrogate constructed by deep neural network usually performs better than the interpolation based one. The search results with the surrogate in place of the simulation system have error margin less than one percent. The rank correlation coefficients of the surrogates are better than 99 percent.

2 MATERIAL AND METHOD

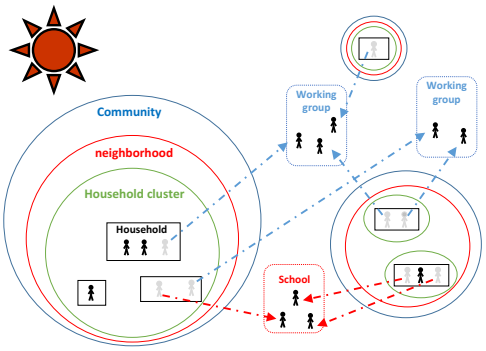
Our approach to search for the optimal vaccination strategy belongs to the general category of simulation-based optimization (Gosavi, 2015). We first introduce our simulation software, followed by a description of our optimization procedure.

We use the simulation software developed in our laboratory (Tsai et al., 2010). Below is a brief description of how the simulation works. It is a stochastic discrete time agent-based model. The mock population of the model is constructed according to national demographics from Taiwan Census 2000 data (http://eng.stat.gov.tw/). The connection between any two individuals indicates the possibility of daily and relatively close contact which could result in a successful transmission of the flu virus. An important virus-dependent parameter is the transmission probability, denoted by p_{trans}. It is the probability that an effective contact results in an infection. An individual can be in one of the following four states, susceptible(S), exposed(E), infectious(I), and recovered(R). when an effective contact happens between an individual in the state S and an individual in the state I, the susceptible individual will become exposed with probability p_{trans}. According to the disease natural history, an exposed individual becomes infectious and an infectious individual becomes recovered, in our setting the average incubation period is 1.9 days and the average infectious period is 4.1 days (Germann et al., 2006). A contact group is a daily close association of individuals, where every member has a contact probability to have effective contact with all other members in the same group. There are eleven such contact groups in the model, they are communities, neighborhood, household cluster, household, workgroup, high school, middle school, elementary school, daycare center, kindergarten, and playgroup (Chang et al., 2015). The population size of Taiwan is about 22.12 million. There are about 1.72 million *preschool children* (0-5 years old), about 2.36 million *elementary school children* (6-12 years old), about 0.99 million *middle school children* (13-15 years old), about 0.97 million *high school children* (16-18 years old), about 3.86 million *young adults* (19-29 years old), about 10.28 million *adults* (30-64 years old), and about 1.94 million *elders* (65+ years old).

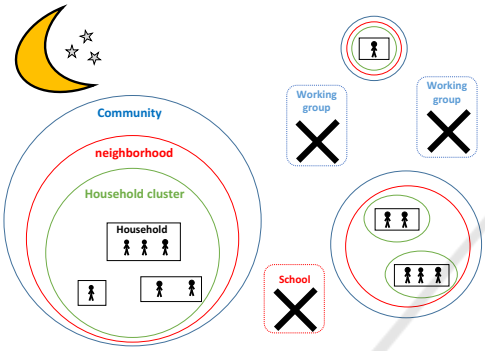
Each individual can belong to several contact groups simultaneously at any time. The duration of a simulation run is set at 365 days. Each day has two 12-hour periods, daytime and nighttime respectively. Behaviors of individuals are depicted in Figure 1. During daytime, contact occurs in all contact group. School aged children go to schools. There are around 7.8% school aged children do not go to school in Taiwan. They stay home in our simulation. Preschool children go to a daycare center, kindergarten or playgroup. Young adults and adults go to workgroup. In the nighttime, contact occurs only in communities, neighborhoods, household clusters, and household.

The model parameters are similar to ones in a study by (Germann et al., 2006), with modifications to fit Taiwan situation better with the help of study outcome in contact diary study (Fu et al., 2012). We recall that the stochastic variation of the simulation system is reported to be around 0.2 percent (Tsai et al., 2010).

In this paper, our setting is similar to our previous study about surrogates functions (Jian et al., 2016):the p_trans is set at 0.1, the vaccine is available 30 days after the index case occurred, total 2.5 million of doses are applied to different age groups according to the priority.. However, we only focus on the case that the vaccine efficacy, VEi and VEs, are fixed at 0.9 (Basta et al., 2008). The vaccine is available 30 days after the index case occurred, total available vaccine are all



(a) Contact behavior in daytime.



(b) Contact behavior in nighttime.

Figure 1: Contact behavior.

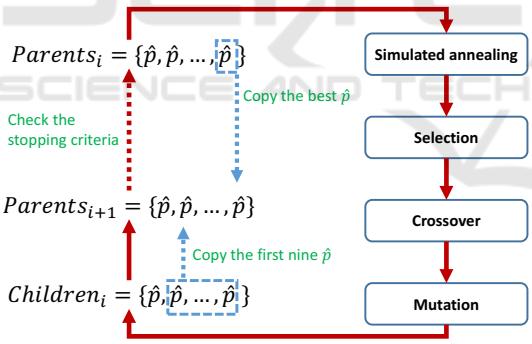


Figure 2: Process of HSAGA.

administered to individuals.

A vaccination priority is denoted by $p = (x_1, x_2, \ldots, x_7)$, where x_i is the amount of vaccine for the age group i and it is less than the population of age group i. We use p_α to denote vectors with only those value of x_α are nonzero and α is a set with numberings of each age group. For example $p_{2,3}$ denote the set of vaccination priority with non-zero entries for age group 2 and age group 3 and $p_{i,j,k}$ to denote the set of vectors with 3 non-zero entries. Let $p = \vec{0}$ denote the baseline case with no vaccination. We use $Sim(p)$ to denote the number of infected cases reported by the simulation program with vaccination priority p.

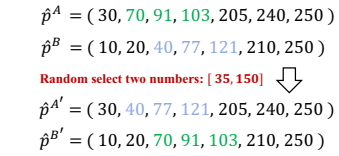


Figure 3: Crossover of HSAGA.

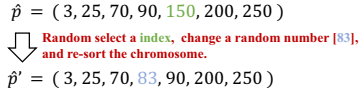


Figure 4: Mutation of HSAGA.

We use the genetic algorithm to search for the optimal vaccine strategy. A simulated annealing step is introduced to speedup the process. In the hybrid simulated annealing genetic algorithm (HSAGA), the population is consists of vaccine allocations represented in prefix sum format. That is $p = (20, 50, 50, 70, 20, 30, 10)$ can be rewritten as $\hat{p} = (20, 70, 120, 190, 210, 240, 250)$, since the total amount of vaccine is always 2.5 millions the last coordinate can be dropped. The population size is ten, and each iteration begins with simulated annealing step to perturb each candidate, followed by the selection, crossover, and mutation steps. Figure 2 is the flow-chart of the process. For a given allocation, we carried out 5 simulation runs, and the fitness score is the average of the values of the objective function of each run. The best solution of the previous generation and the first nine solutions for this generation become the candidates of next generation. At the beginning of each iteration, we carry out a simulated annealing step for each candidate. It is a temperature controlled mutation, i.e., we mutate each candidate according to the temperature (that is the number of iterations up to the point in our case). The process stops at 200 iterations and the early stop condition is that five consecutive iterations consist of the same candidates. Given two genes (vaccine priorities), the crossover operation is the following: Randomly generate a pair of numbers g_1, g_2 where $0 \leq g_1 \leq g_2 \leq 250$, if the interval $[g_1, g_2]$ covers the same number of chromosomes in both genes, then we exchange the covered part. The segment of chromosomes x_i, x_j is covered by interval $[g_1, g_2]$ if and only if $x_{i-1} \leq g_2 \leq x_i$ and $x_j \leq g_2 \leq x_{j+1}$. Figure 3 is an example of the crossover operation. We randomly increase g_1 or decrease g_2 if a direct exchange is invalid, that is the length of covered segments differ. A more detailed description can be found in (Jian et al., 2016).

The mutation operation is defined as following: Randomly pick index i and randomly generate a number x, replace x_i with x and sort the resultant sequence. Figure 4 is an illustration of the mutation operation.

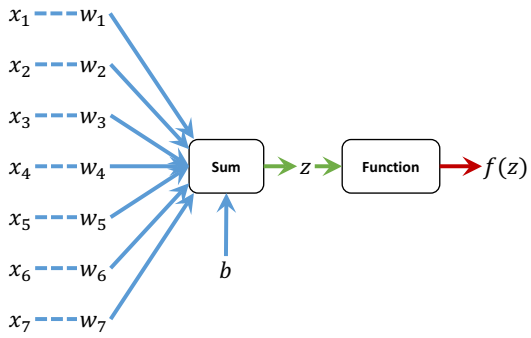


Figure 5: The model of $Sur_1^D(p)$.

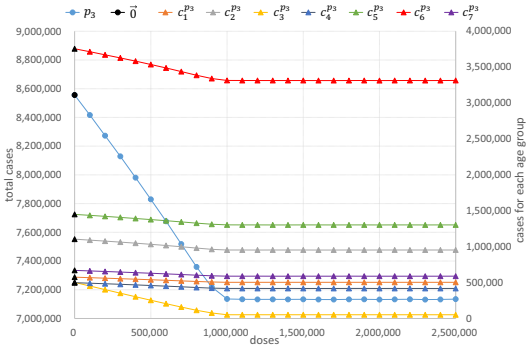


Figure 6: $Sim(p_3)$ with $p_3 \in P_t$.

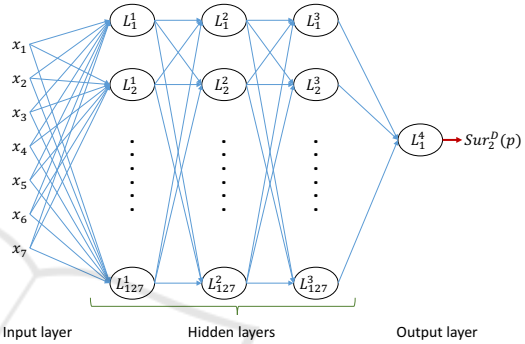


Figure 7: The model of $Sur_2^D(p)$.

2.1 Surrogates with Deep Learning

Deep learning gained a lot of attention with a few highly publicized success. It is a branch of machine learning, and one of its basic ingredients is the artificial neural network. The word deep referring to the fact that the model is built with multilayer perceptrons(MLP). Each perceptron has a transformation function to produce output to next perceptrons with the inputs from connected perceptrons at the previous stage. During the training phase, the prediction error is determined by the loss function, and the error triggering a weight adjustment procedure, backpropagation is the most commonly applied method.

In our research, we use deep neuron networks (DNN) as the model of the surrogate function. In this study, we use Keras on Theano running on Nvidia GeForce GTX 1080 Graphics Card (Keras, 2015).

The surrogate function takes the vaccine allocation of seven age groups, p, as the input and the output is the total number of infected cases.

We first applied single perceptron model with the linear function to check if there exist good linear surrogates. The single perceptron model is shown in Figure 5, in which w_i denotes the weight, and b is the bias. We let the activation function to be a linear function, i.e., $f(z) = z$ and the output of the single perceptron model is denoted by Sur_1^D

$$Sur_1^D(p) = f(z) = z = \sum_{i=1}^{7} (w_i x_i) + b \qquad (1)$$

As shown in Figure 6, the slope approaching zero when the vaccine allocation is approaching the size of the population of that age group. As expected, we can see that the number of infected cases of other age groups is also affected by the amount allocated to one specific age group.

We next move on to the deep neural network with nonlinear activation function. The architecture is shown in Figure 7. The output of the deep neural network is denoted by $Sur_2^D(p)$. It is a fully connected architecture, that is the outputs of every neuron in this layer are inputs for every neuron in next layer. We add three hidden layers, and each layer has $\sum_{i=1}^{7} C_i^7 = 127$ perceptrons, which corresponding to the number of all the combinations of the seven age groups. The activation function is exponential linear units (ELUs), it can handle non-linear relations and outperform the traditional rectified linear unit(ReLU)(Clevert et al., 2015).

2.2 Surrogates with Interpolation

We compare the surrogates learnt by neural networks with the interpolation based ones we constructed before. We first give a brief description of the previous work, and the details can be found in (Jian et al., 2016).

To apply interpolation, we need a set of reference points, denoted by P_t, the values of these points are the simulation results. In other words, points in p_t are entries of the lookup table. If $p \in P_t$ the result, $Sim(p)$, is the total number of infected cases by simulation. If $p \notin P_t$, then $Sim(p)$ denote the estimated total number of infected cases by interpolation. We first sampled 26 points for each age group, there are 182 points in total. The 26 points are evenly spaced up to 2.5 mil-

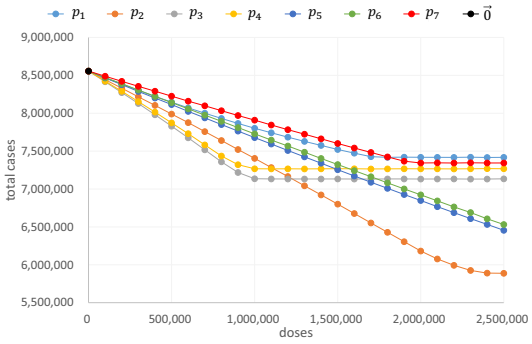


Figure 8: $Sim(p_i)$ with $p_i \in P_t$.

lion doses of vaccine. Figure 8 shows the value of each point. We then define the effect of introducing strategy p_i, denoted by $\Delta(p_i)$ as following:

$$\Delta(p_i) = Sim(p_i) - Sim(\vec{0}) \quad (2)$$

To approximate the effect of a point p, we add the effect of each component and denote it by $Sur_1^I(p)$:

$$Sur_1^I(p) = Sim(\vec{0}) + \sum_{i=1}^{7} \Delta(p_i) \quad (3)$$

If the value of p_i is not sampled, the linear interpolation is applied to estimate $\Delta(p_i)$. As expected, this simple surrogate does not capture the interaction between age groups well. We build a lookup table with entries in the form of $p_{i,j}$ to capture the interactions between age groups. For each age group i, five evenly spaced values are determined. Given two age groups, we take one value from each group to form a vaccination strategy and carry out the simulation. There are total $21 \times 5 \times 5 = 525$ such points. For a point $p = (x_1, x_2, ..., x_7)$, we slightly abuse the notation and use $p_{j,k}$ denote the point with the same value as p in j^{th} and k^{th} dimensions and all other dimensions are zero. The adjustment term for the interaction between dimension j and k, denoted by $\delta(p_{j,k})$, is defined as following:

$$\delta(p_{j,k}) = Sim(p_{j,k}) - Sur_1^I(p_{j,k}) \quad (4)$$

When $p_{j,k}$ is not sampled, a bilinear interpolation is applied. The surrogate with the two age group interaction adjustment, denoted by $Sur_2^I(p)$, is defined as following:

$$Sur_2^I(p) = Sim(\vec{0}) + \sum_{i=1}^{7} \Delta(p_i) + \sum_{j=1}^{6} \sum_{k=j+1}^{7} \delta(p_{j,k}) \quad (5)$$

3 RESULTS

We collected two sets of points in our previous study. First, the set of base points, p_t, which are the points serve as the sampled points while developing interpolation based surrogate. Second, the set of points evaluated during the execution of HSAGA and we denote the set by P_h. For this study, we further evaluate a set of points, denoted by P_q, which are points have 3 age groups assigned none zero entries. There are $C_3^7 = 36$ combinations and for each dimension, we evenly sampled 4 points up to the population size of that age group. In other words, the increment is $1/4$ of the size of the age group. We also limit the total amount is no more than 4 million doses and each age group gets at most 2.5 million. We note that because of the choice of increment $P_t \bigcap P_q = \emptyset$. There size of P_t is 707, P_h is 988, and P_q is 1557. Our training and testing data are drawn from these three sets. We set epoch to be 10 thousand, mini-batch to be 10 and we use mean absolute error (MAE) as our loss function and Nadam as optimizer (Dozat, 2015).

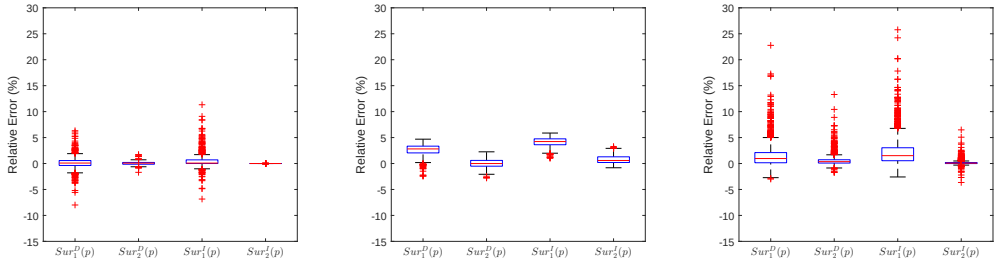
We compare surrogates learnt by DNN with interpolation based surrogates. There are several settings. We always use D_a to denote the training set and D_b to denote the testing set. The interpolation based surrogate does not have the training phase, only testing set matters for their evaluation. We use the relative error between the output of surrogate and the simulation result and use box plot to visualize the results.

In Figure 9(a), there is no error for $Sur_2^I(p)$ because the testing data is P_t which are the reference base for interpolation. Similarly, the error for $Sur_1^I(p)$ is from points with pattern $p_{j,k}$. From the fact that $Sur_1^D(p)$ is the result of a thorough training phase and that it has much larger error comparing with $Sur_2^D(p)$, it is safe to say that the relation between vaccine allocation and the total number of infected case reduced is not a simple linear one.

In Figure 9(b), we note that P_h contains points with many non-zero dimensions because the genetic algorithm starts with random points and gradually converge to points concentrating on vaccinating school children. We can see that $Sur_1^D(p)$ and $Sur_1^I(p)$ obviously over-estimate the value, although the spreading patterns are more or less similar in all four cases.

In Figure 9(c), the testing data P_q contains points with 3 non-zero entries. Compared with Figure 9(a) the over-estimating phenomenon is even more obvious, even $Sur_2^I(p)$ can not remedy the fact that the interaction among three age groups is not captured.

We can see that $Sur_2^I(p)$ is outperformed by $Sur_2^D(P)$ in Figure 9(a) and 9(c) . The reason is that



(a) $D_a = P_t, D_b = P_t$ (b) $D_a = P_t, D_b = P_h$ (c) $D_a = P_t, D_b = P_q$

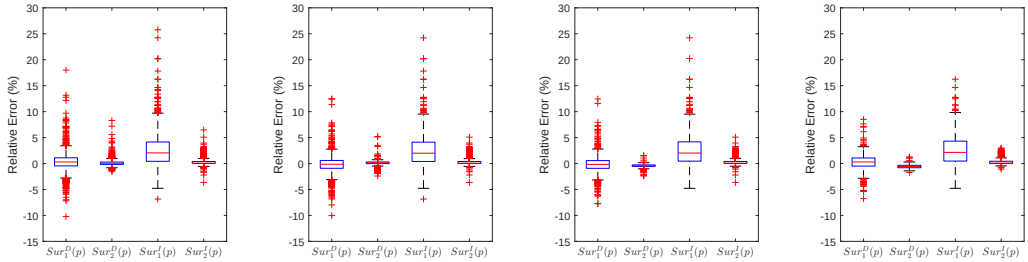
(d) $D_a = P_t \bigcup P_h, D_b = P_h$ (e) $D_a = P_t \bigcup P_q, D_b = P_q$ (f) $D_a = D_b = P_t \bigcup P_h \bigcup P_q$

Figure 9: Box plot with different surrogates.

Table 1: Detail data of Figure 9.

	surrogate	avg of abs	max of abs	avg	SD	Q_1	Q_2	Q_3	IQR
Figure 9(a)	$Sur_1^D(p)$	0.80	8.00	0.11	1.25	-0.38	0.09	0.57	0.95
	$Sur_2^D(p)$	0.23	1.71	0.05	0.31	-0.14	0.08	0.21	0.34
	$Sur_1^I(p)$	0.77	11.32	0.48	1.43	0.00	0.08	0.70	0.70
	$Sur_2^I(p)$	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.00
Figure 9(b)	$Sur_1^D(p)$	2.65	4.69	2.61	1.06	2.04	2.82	3.35	1.30
	$Sur_2^D(p)$	0.68	2.79	0.01	0.88	-0.52	-0.01	0.60	1.11
	$Sur_1^I(p)$	4.16	5.87	4.16	0.84	3.62	4.23	4.75	1.13
	$Sur_2^I(p)$	0.86	3.28	0.80	0.83	0.20	0.58	1.29	1.09
Figure 9(c)	$Sur_1^D(p)$	1.76	22.77	1.43	2.28	0.15	0.96	2.11	1.96
	$Sur_2^D(p)$	0.66	13.30	0.56	1.00	0.09	0.36	0.73	0.64
	$Sur_1^I(p)$	2.42	25.77	2.23	2.84	0.54	1.52	3.04	2.50
	$Sur_2^I(p)$	0.22	6.48	0.15	0.44	-0.01	0.08	0.19	0.20
Figure 9(d)	$Sur_1^D(p)$	1.00	5.81	0.02	1.30	-0.66	0.17	0.97	1.63
	$Sur_2^D(p)$	0.19	0.73	-0.14	0.19	-0.27	-0.14	-0.02	0.25
	$Sur_1^I(p)$	4.16	5.87	4.16	0.84	3.62	4.23	4.75	1.13
	$Sur_2^I(p)$	0.86	3.28	0.80	0.83	0.20	0.58	1.29	1.09
Figure 9(e)	$Sur_1^D(p)$	1.20	17.81	0.21	1.85	-0.62	0.13	0.88	1.50
	$Sur_2^D(p)$	0.29	3.56	-0.24	0.30	-0.37	-0.24	-0.11	0.26
	$Sur_1^I(p)$	2.42	25.77	2.23	2.84	0.54	1.52	3.04	2.50
	$Sur_2^I(p)$	0.22	6.48	0.15	0.44	-0.01	0.08	0.19	0.20
Figure 9(f)	$Sur_1^D(p)$	1.30	16.21	-0.85	1.60	-1.58	-0.77	-0.04	1.54
	$Sur_2^D(p)$	0.18	3.88	-0.10	0.25	-0.21	-0.08	0.04	0.24
	$Sur_1^I(p)$	2.59	25.77	2.44	2.50	0.44	2.09	4.15	3.72
	$Sur_2^I(p)$	0.37	6.48	0.31	0.64	0.00	0.07	0.38	0.38

the training data and testing data are in a different category and $Sur_2^I(P)$ is designed to work with those special categories well. In Figure 9(b) the testing set contains more randomly sampled data, we see $Sur_2^D(p)$



(a) 20%, 80% (b) 40%, 60% (c) 60%, 40% (d) 80%, 20%

Figure 10: Box plot which cutting sample data $P_t \bigcup P_h \bigcup P_q$ by different percentage for D_a and D_b.

Table 2: Detail data of Figure 10.

	surrogate	avg of abs	max of abs	avg	SD	Q_1	Q_2	Q_3	IQR
Figure 10(a)	$Sur_1^D(p)$	1.14	18.00	0.30	1.64	-0.46	0.30	1.10	1.57
	$Sur_2^D(p)$	0.30	8.29	0.09	0.49	-0.18	0.04	0.82	0.46
	$Sur_1^I(p)$	2.60	25.77	2.45	2.55	0.43	2.05	4.15	3.72
	$Sur_2^I(p)$	0.38	6.48	0.32	0.65	0.00	0.07	0.39	0.39
Figure 10(b)	$Sur_1^D(p)$	1.06	12.48	-0.21	1.57	-0.92	-0.15	0.59	1.51
	$Sur_2^D(p)$	0.26	5.20	0.12	0.39	-0.02	0.14	0.30	0.32
	$Sur_1^I(p)$	2.54	24.22	2.39	2.49	0.39	1.99	4.11	3.72
	$Sur_2^I(p)$	0.38	5.07	0.32	0.65	0.00	0.06	0.38	0.38
Figure 10(c)	$Sur_1^D(p)$	1.08	12.44	-0.22	1.60	-0.93	-0.18	0.57	1.50
	$Sur_2^D(p)$	0.44	2.40	-0.43	0.27	-0.57	-0.42	-0.27	0.30
	$Sur_1^I(p)$	2.62	24.22	2.47	2.58	0.46	2.01	4.18	3.72
	$Sur_2^I(p)$	0.39	5.07	0.33	0.66	0.00	0.07	0.40	0.40
Figure 10(d)	$Sur_1^D(p)$	1.13	8.51	0.24	1.55	-0.50	0.30	1.07	1.57
	$Sur_2^D(p)$	0.59	1.73	-0.57	0.33	-0.81	-0.58	-0.35	0.46
	$Sur_1^I(p)$	2.70	16.24	2.55	2.53	0.47	2.14	4.31	3.84
	$Sur_2^I(p)$	0.38	3.00	0.33	0.61	0.00	0.08	0.44	0.44

performs better. In the next few experiments, we allow training set to contain the testing set. We are fully aware that training set and testing set should be disjoint in general. But here we want to demonstrate the advantage of the machine learning approach, that is by providing proper training set the performance can be enhanced greatly. As expected, in Figure 9(d) and Figure 9(f) we can see $Sur_2^D(p)$ learned a better approximation function and outperforms $Sur_2^I(p)$. In Figure 9(e), $Sur_2^I(p)$ performs better and we suspect that the testing set, points with three non-zero elements, is very close to the table lookup entries, points with two non-zero elements. And this particular phenomenon deserve further investigation.

A proper evaluation should have the disjoint training set and testing set. We take the union of P_t, P_h and P_q as the sample set. And then partition the whole set into training and testing set. The result is shown in Figure 10, the percentage on the left is the portion of the training set and left is the testing set. Observing the quartile,Q_1 andQ_3, of $Sur_2^D(p)$ and $Sur_2^I(p)$,

Table 3: The best allocation of *HSAGA* with each fitness function.

F	C	I	N	p ($\times 10^4$ doses)		
				ES	MS	HS
$Sim(p)$	4.99	72	988	97	79	74
$Sur_2^D(p)$	4.98	75	1020	100	81	69
$Sur_2^I(p)$	4.99	87	1135	96	78	76

' F ': fitness function
' C ': total cases ($\times 10^6$)
' I ': total iterations
' N ': total allocations
' ES ': *elementary school children*
' MS ': *middle school children*
' HS ': *high school children*

we can see the effect of learning. Also, the width of the spreading pattern decreases as the training data increases. We include all the numerical data for the plot in Table 2. There are two columns "avg" and "avg of abs", the former is the average of the relative error and the latter is the average of the absolute value of

relative error.

We then put $Sur_2^D(p)$ to the application test, and use it as the fitness function in HSAGA to search for the appropriate vaccine strategy. The training set for the function is $P_t \bigcup P_h \bigcup P_q$. The result is shown in Table 3. Three methods produce similar results and the conclusion confirms to previously reported studies: that the best strategy is to vaccinate school children.

To visualize the vaccine allocations and to facilitate further exploration, a method to encode the allocation by gray level was developed in (Jian et al., 2016). We briefly recap the method. There are two encoding schemes,*volume scheme* and *ratio scheme*. For volume scheme, the color white is to denote zero doses and black for 2.5 million doses. Let x_i be the number of doses for age group i, the gray level is computed by the following equation:

$$g_i^{volume} = 255 \times [1 - \frac{x_i}{2.5 \text{ million doses}}] \quad (6)$$

For the ratio scheme, the color white to denote zero percent of the age group i vaccinated and black hundred percent and we used $\#AG_i$ to denote the population of age group i. The gray level is computed by following equation:

$$g_i^{ratio} = 255 \times [1 - \frac{x_i}{\#AG_i}] \quad (7)$$

After each age group is assigned a gray level according to the equation above, we use a line segment with that gray level to represent vaccination level of each age group, as shown in the top half of Figure 11. The allocation is then represented by stacking the seven line segment vertically (in the middle part of Figure 11, we put the line segment horizontally).

For a set of ordered allocations, the line segment for each allocation is stitched together according to the ordering. The sequence of allocations is sorted from left to right where the better allocations are on the right hand side. As shown in Table 4, the visual effect of concentration on school children is obvious.

For genetic algorithms, the rank preserving surrogates are preferred. One metric to measure the fidelity of surrogates is rank correlation coefficient (r_s) (Loshchilov et al., 2010):

$$r_s = 1 - \frac{6 \times \sum_{i=1}^{N} (R_A[i] - R_B[i])^2}{N(N^2-1)} \quad (8)$$

We compute the rank correlation coefficient for all surrogates with all sampled points in the list. The coefficients of $Sur_2^D(p)$ are all above 99 percentage and the numbers are shown in Table 5, the left column indicate the domain of elements and the $Sur_2^D(p)$ is trained with $P_a = P_t \bigcup P_h \bigcup P_q$. $Sur_2^D(p)$ has the best coefficient except the case where all elements are from P_t.

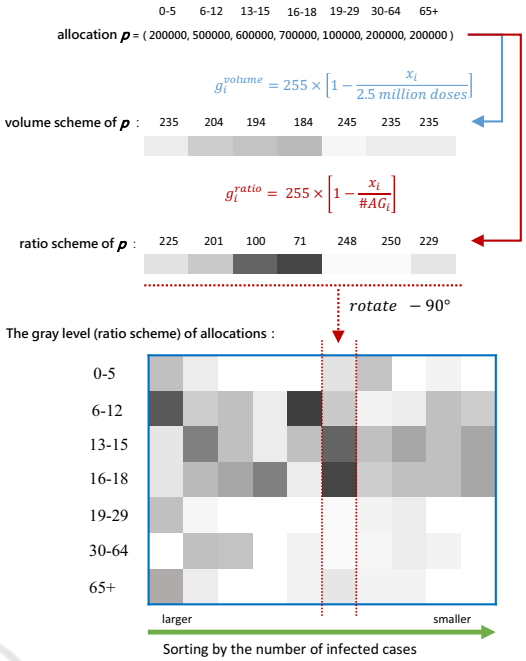


Figure 11: The gray level(Jian et al., 2016).

Table 4: The gray level of total allocations of *HSAGA* with each fitness function.

function	allocations
$Sim(p)$	
$Sur_2^D(p)$	
$Sur_2^I(p)$	

4 CONCLUSION AND DISCUSSION

We explore the feasibility of using machine learning approach to constructing surrogates as the cost function for optimization schemes. The training data is generated by simulation system. It is natural to suspect that the cost of generating enough training points would be higher than using simulation as the cost function during the optimization process. We thus try to utilize the data points recorded by previous study and discover that those data points can be reused to produce good surrogate by the deep neural network.

Table 5: Rank correlation coefficient.

	r_s			
	$Sur_1^D(p)$	$Sur_2^D(p)$	$Sur_1^I(p)$	$Sur_2^I(p)$
P_t	0.9974	0.9999	0.9990	0.9999
P_h	0.9155	0.9972	0.9516	0.9611
P_q	0.9958	0.9999	0.9999	0.9999
P_a	0.9913	0.9998	0.9902	0.9976

' P_a ': $P_t \bigcup P_h \bigcup P_q$

It is clearly demonstrated that linear approximations,$Sur_1^I(p)$ and $Sur_1^D(p)$, are much less accurate than more complicated approximations, $Sur_2^I(p)$ and $Sur_2^D(p)$. It illustrated that there are interesting interactions among age groups. We also confirmed that if the training set and testing set are very different, the performance of $Sur_2^D(p)$ is less impressive than $Sur_2^I(p)$. However, when the training set does contain similar points as testing set, $Sur_2^D(p)$ outperforms $Sur_2^I(p)$. It is more accurate and has higher rank coefficient correlation.

Our results confirm the finding of previous studies that school children should be vaccinated with high priority. One obvious future direction is to use machine learning to explore the vast landscape of scenarios with various objective functions and constraints. For example, the infectiousness of the virus strand might vary, the available date of vaccine may not be known in advance, and other mitigation strategies such as antiviral treatment and school closure might vary. The objective function can vary too. Instead of minimizing infected cases, one might want to minimize economic cost (Meltzer et al., 1999). Currently, we the only variables are the amounts of vaccine allocated to different age groups. More parameters will be included as input and we plan to try the convolutional neural network and the recurrent neural network in the future. We hope not only we can construct accurate surrogates with more parameters, but also can gain insight about the delicate interaction between model parameters and outcome by studying the neural networks.

Finally, we envision that an autonomous software searches through the huge scenario space with the help of surrogate function and adaptively executes simulation program to revise the surrogate function to produce higher fidelity surrogate and better search results by applying reinforcement learning methods.

ACKNOWLEDGEMENTS

We thank anonymous reviewers for their suggestions. The research is partially funded by the grant of "MOST105-2221-E-001-034", "MOST104-2221-E-001-021-MY3", and "Multidisciplinary Health Cloud Research Program: Technology Development and Application of Big Health Data. Academia Sinica, Taipei, Taiwan".

REFERENCES

Basta, N. E., Halloran, M. E., Matrajt, L., and Longini, I. M. (2008). Estimating influenza vaccine efficacy from challenge and community-based study data. *American Journal of Epidemiology*, 168(12):1343–1352.

Chang, H.-J., Chuang, J.-H., Fu, Y.-C., Hsu, T.-S., Hsueh, C.-W., Tsai, S.-C., and Wang, D.-W. (2015). The impact of household structures on pandemic influenza vaccination priority. In *Proceedings of the 5th International Conference on Simulation and Modeling Methodologies, Technologies and Applications - Volume 1: SIMULTECH,*, pages 482–487.

Chao, D. L., Halloran, M. E., Obenchain, V. J., and Longini, Jr, I. M. (2010). Flute, a publicly available stochastic influenza epidemic simulation model. *PLOS Computational Biology*, 6(1):1–8.

Clevert, D., Unterthiner, T., and Hochreiter, S. (2015). Fast and accurate deep network learning by exponential linear units (elus). *CoRR*, abs/1511.07289.

Dozat, T. (2015). Incorporating nesterov momentum into adam. http://cs229.stanford.edu/proj2015/054_report.pdf.

Fu, Y.-c., Wang, D.-W., and Chuang, J.-H. (2012). Representative contact diaries for modeling the spread of infectious diseases in taiwan. *PLoS ONE*, 7(10):1–7.

Germann, T. C., Kadau, K., Longini, I. M., and Macken, C. A. (2006). Mitigation strategies for pandemic influenza in the United States. *Proceedings of the National Academy of Sciences*, 103(15):5935–5940.

Gosavi, A. (2015). Simulation-based optimization. *parametric optimization techniques and reinforcement learning*.

Jian, Z.-D., Hsu, T.-S., and Wang, D.-W. (2016). Searching vaccination strategy with surrogate-assisted evolutionary computing. In *Proceedings of the 6th International Conference on Simulation and Modeling Methodologies, Technologies and Applications - Volume 1: SIMULTECH,*, pages 56–63.

Jin, Y. (2011). Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 2(1):61–70.

J.J. Grefenstette, J. F. (1985). Genetic search with approximate fitness evaluations. In *International Conference on Genetic Algorithms and Their Applications*, pages 112–120.

Keras (2015). Keras documentation. https://keras.io/.

Loshchilov, I., Schoenauer, M., and Sebag, M. (2010). *Parallel Problem Solving from Nature, PPSN XI: 11th International Conference, Kraków, Poland,*

September 11-15, 2010, Proceedings, Part I, chapter Comparison-Based Optimizers Need Comparison-Based Surrogates, pages 364–373. Springer Berlin Heidelberg, Berlin, Heidelberg.

Meltzer, M. I., Cox, N. J., and Fukuda, K. (1999). The economic impact of pandemic influenza in the United States: priorities for intervention. *Emerging Infect. Dis.*, 5(5):659–671.

Tsai, M.-T., Chern, T.-C., Chuang, J.-H., Hsueh, C.-W., Kuo, H.-S., Liau, C.-J., Riley, S., Shen, B.-J., Shen, C.-H., Wang, D.-W., and Hsu, T.-S. (2010). Efficient simulation of the spatial transmission dynamics of influenza. *PLOS ONE*, 5(11):1–8.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS