AIChE WILEY InterScience DISCOVER SOMETHING GREAT

PROCESS SYSTEMS ENGINEERING

Nonlinear Experimental Design using Bayesian Regularized Neural Networks

Matthew C. Coleman
Dept. of Chemical Engineering and Material Science, University of California, One Shields Ave, Davis, CA 95616

David E. Block
Dept. of Viticulture and Enology, and Dept. of Chemical Engineering and Material Science, University of California, One Shields Ave, Davis, CA 95616

DOI 10.1002/aic.11175
Published online April 16, 2007 in Wiley InterScience (www.interscience.wiley.com).

Novel criteria for designing experiments for nonlinear processes are presented. These criteria improve on a previous methodology in that they can be used to suggest a batch of new experiments to perform (as opposed to a single new experiment) and are also optimized for discovering improved optima of the system response. This is accomplished by using information theoretic criterion, which also heuristically penalize experiments that are likely to result in low (nonoptimal) results. While the methods may be applied to any type of nonlinear-nonparametric model (radial basis functions and generalized linear regression), they are here exclusively considered in conjunction with Bayesian regularized feedforward neural networks. A focus on the application of rapid process development, and how to use repeated experiments to optimize the training procedures of Bayesian regularized neural networks is shown. The presented methods are applied to three case studies. The first two case studies involve simulations of one and two-dimensional (2-D) nonlinear regression problems. The third case study involves real historical data from bench-scale fermentations generated in our laboratory. It is shown that using the presented criteria to design new experiments can greatly increase a feedforward neural network's ability to predict global optima.
 AIChE J, 53: 1496–1509, 2007
Keywords: nonlinear experimental design, Bayesian regularization, rapid process development, fermentation

Introduction

The design and optimization of new production processes, and the improvement on existing ones is a critical step for most industrial products. From manufacturing automobiles to the production of therapeutic drugs there are numerous process variables that must be decided on before production begins. There are many popular methods available for designing experiments to determine optimal values for such process variables.[1–3] These traditional methods of designing experiments (for example, factorial or central composite designs) have of course been shown to be useful in optimizing a wide variety of systems. However, such designs are often limited when the system diverges systematically from linear behavior. Fixed form models (for example, linear, polynomials) may not be sufficient to explain the available data. In addition, gradient based methods of experimental optimization (for example, response surface methodology or sequential fractional factorial designs)[4,5] may get caught in local optima. Using more efficient experimental designs will hope-

Correspondence concerning this article should be addressed to D. E. Block at deblock@ucdavis.edu.

fully lead to optimal performance in fewer experiments, thus, saving time and money.

The goal of this work is to design experiments in a manner that will determine optimal values of process variables of nonlinear systems in as few experiments as possible. The systems under consideration have a single response (for examples, the product yield from biochemical processes), and have one to several independent variables over which the practitioner has some control (for examples, the temperature and pH of a reactor). The design of new experiments should use all available process data that may have been gathered from a range of sources from simple traditional experimental designs (for example, two-level factorial designs) to historical data sets, that have been generated from random sets of processing inputs. While the criteria developed in this work for identifying the best set of next experiments, can be used in conjunction with any nonlinear model, we exclusively consider their use in conjunction with Bayesian regularized feedforward neural networks.[6–10] Furthermore, we show how repeated experiments (which are typically found in process databases for new processes) can be utilized to improve the performance of Bayesian regularization.

Some methods for designing experiments for nonlinear systems already exist. Atkinson and Donev[2] present methods that efficiently design experiments for nonlinear systems, where the form of the nonlinear model is known. Chaloner and Verdinelli,[11] as well as Loredo and Chernoff[12] present fully Bayesian methods of experimental design from a decision theoretic perspective; however, these methods also tend to rely on more structurally defined models. Here we are concerned with systems where no specific structure is known beforehand, hence, our decision to use neural networks, which have been shown to be effective at modeling arbitrarily complex systems. While the methods that we present here do include some heuristic strategies, and are not fully Bayesian, they are computationally much simpler making them more attractive from a practitioner's point of view.

MacKay[13] first suggested criteria that could be used to design experiments using neural network models. MacKay's work was based on the work of Fedorov,[1] and uses information theoretic criteria to quantify the uncertainty in model parameters. By choosing experiments that minimize the expected uncertainty of the model parameters, data are collected that best update the model. MacKay[13] utilized these criteria in conjunction with Bayesian regularized neural networks. Cohn [14] and Cohn et al. [15] generalized these criteria for other types of neural network training methods, as well as mixtures of Gaussian and generalized linear models. Similar methods have been applied to process development,[16] however, none of these approaches address the necessity of quickly determining system maxima or minima in a process development setting. Other similar approaches have been proposed to design experiments using neural network models,[17,18] although all of these approaches require evaluation of complex objective functions, and none allow for the inclusion of information from repeated experiments.

There are three important additions to previous work that we present. First, we describe how traditionally designed experiments can be used to best train Bayesian regularized neural networks. In such experimental designs, there are always repeated experiments that are used to estimate the noise level of the system under study. This estimated noise level can be used to form an empirical Bayesian prior. This prior can be used in conjunction with the training algorithm for Bayesian regularized neural networks[8,10] to improve the generalization capacities of the developed model. The use of such informative priors has been shown to improve parameter estimates of various types of models,[19,20] but informative priors over noise parameters have yet to be applied to training neural network models. Second, we introduce a criterion that can be used to suggest multiple new experiments to perform simultaneously. Performing several experiments at one time, to optimize the information gained about the system is an important task, as the parallel experiments will allow all available equipment to be utilized. Traditional methods, such as D-optimal design can be used to efficiently design multiple experiments, however, they are not able to make use of arbitrarily complex models, such as the Bayesian regularized NN models in this article. Third, we introduce a heuristic approach for modifying these criteria for choosing new experiments in order to more efficiently find new optima of the system.

A theory section reviews the basics of Bayesian regularized neural networks and introduces the criteria to be used in this work for choosing new experiments. These criteria are as follows: total information gain (TIG), batch-total information gain (BTIG), relative information gain (RIG), and batch-relative information gain (BRIG). Only the TIG has previously been discussed.[13] Three case studies are then used to evaluate the introduced criteria.

Materials and Methods

Fermentation processing

The third case study uses a historical *Escherichia coli* fermentation database that was generated using a strain obtained from Drs. William Bentley and Govind Rao at the University of Maryland, College Park and the University of Maryland, Baltimore County, respectively. This *E. coli* strain, JM 105 (F' *Δlac-pro thi strA endA sbcB15 hspR4 tra36 pro* AB^+ $lacI^q$ *-ZΔM15*), bears the plasmid [pBAD-GFP::CAT][21]. The GFP and CAT reporters each possess a ribosome-binding site; however, both are under the control of the pBAD promoter of the *ara*BAD (arabinose operon). *E. coli* JM105 [pBAD-GFP::CAT] was induced for expression of green fluorescent protein (GFP) by the addition of appropriate amounts of arabinose.[20,22–24]

All fed-batch experiments were carried out in eight BioFlo 3000 fermentors, each with a 5 L working volume (New Brunswick Scientific, N J). All inocula were grown in an Innova 4000 shaker (New Brunswick Scientific, N J) at 37°C and 200 rpm, and were added to the fermentors by gravity. Media for the fermentations (4 L initially per fermentor) was prepared by combining yeast extract (Amberex 695, Universal Flavors, Juneau, WI, Tastone 900, Universal Flavors, Juneau, WI, or Difco Yeast Extract, Becton-Dickinson, Sparks, MD), tryptone (Fisher BioTech, NJ), D(+)-glucose (5 g/L, Sigma Chemical Company, St. Louis, MO), and NaCl (40 g/L, Fisher Scientific, USA). Feed solutions containing 400 g/L D(+)-glucose (Sigma Chemical Company, St. Louis, MO) in deionized water were sterilized separately by autoclaving. The glucose feed was started when the broth reached an optical density reading (600 nm) of 2.0 (0.4 g

Table 1. Process Inputs for Experimental *E. coli* Database in Case Study Three

Input Class	#	Input Name	Range	Description
Fermentation Conditions	1	pH	6.70–7.45	Controlled at set point with acid/base feed
	2	Temperature (°C)	30–37	Controlled at set point with cooling jacket
	3	DO (% saturation)	20–40	Controlled at set point via agitation
Media Variables	4	Yeast Extract Concentration (g/L)	5–17.5	In initial medium
	5	Tryptone Concentration (g/L)	15–32.5	In initial medium
	6	Percent Arabinose (%wt/vol)	0.05–0.20	Used for induction; three times on an hourly basis
	7	Induction Time (hrs)	1–3.5	Time for initial induction following initiation of feed
	8	Yeast Extract Source	Difco, Tastone, Amberex	Source of yeast extract
	9	Feed Strategy (mL/min/g/L)	$\frac{1}{10}, \frac{1}{5}, \frac{2}{5}$	Glucose feed rate set at a ratio proportional to the cell density after reaches 0.8g/L
	10	Antifoam	Before, After	Antifoam added before or after autoclaving fermentors
	11	Antibiotic	No, Yes	Ampicillin added to fermentor
Inoculum Conditions	12	Inoculation Volume (mL)	100–400	Volume used to inoculate 4 L of medium
	13	Inoculation Time (hrs)	6–12	Time that inoculum is allowed to grow prior to use
	14	Cell Bank	1, 2	Origin of initial inoculum

cells/L), and was adjusted every two hours to a rate proportional to the optical density. Induction was achieved by injecting L-arabinose (Sigma Chemical Company, St. Louis, MO) solution, every hour for three hours, beginning either two or four hours after the start of glucose feed. A stock solution of ampicillin (100 mg/mL) was filter sterilized and stored at 4°C; appropriate amounts, based on 4 L media, were added according to the experimental design. The pH was controlled by the addition of 2 N sulfuric acid and 2 N sodium hydroxide. The dissolved oxygen (DO) level inside the fermentor vessel was controlled using agitation, with the minimum and maximum levels of agitation set at 200 and 1,000 rpm, respectively. The experiments were designed in such a manner that no addition of pure oxygen was required. The foam level was maintained by the addition of appropriate amounts (ca. 1 mL) of Antifoam 289 (Sigma Chemical Company, St. Louis, MO). All fermentation inputs along with their operating ranges are shown in Table 1. 45 unique combinations of these fermentation inputs (along with three repeated combinations) were performed. These experiments were chosen based on a randomly generated partial factorial design, where continuous variables had five to seven levels. Additional experimental details can be found in [23–25].

Computation

All computational algorithms were implemented using Matlab v6.5 (The Math Works, Inc., Natick, MA), and run on personal computers (3 GHz, 2 GB RAM). Two Matlab toolboxes were also utilized: the optimization toolbox (The Math Works, Inc., Natick, MA), and the neural network toolbox (The Math Works, Inc., Natick, MA).

Theory

Neural network training

The general task for the NN models used here is to map an output response (y), from a set of input variables ($\mathbf{x}$), given a set of observations. The collected data is denoted as, $D_N = \{\mathbf{x}_i, y_i\}_{i=1}^N$, where there have been a total of N observations, and y_i and $\mathbf{x}_i$ are the ith observations of the output and input values, respectively. A typical NN training algorithm aims to minimize the sum of squared error between neural network predictions, and observed output values by adjusting a vector of neural network parameters ($\mathbf{w}$). The minimization criterion, is, thus

$$S = \sum_{i=1}^{N} \frac{1}{2} (f(\mathbf{x}_i|\mathbf{w}, \mathcal{A}) - y_i)^2 \tag{1}$$

where S is the sum of squared error between the observed output responses and the NN predictions, and $f(\mathbf{x}_i|\mathbf{w},\mathcal{A})$ denotes the NN prediction for the ith input vector given a vector of network parameters $\mathbf{w}$, and a network architecture denoted by $\mathcal{A}$ (for example, one hidden layer with ten nodes).

Regularization training methods add an additional term to the minimization criterion that penalizes NNs that have parameter vectors ($\mathbf{w}$) with a large range of values. When parameter vectors have a large range they create NN output surfaces that are more complex (that is, steeper slopes and more local minima and maxima). This additional term is most often the sum of squared weights $W = 1/2\Sigma_{j=1}^{N_W} w_j^2$, where there are N_W weights in the network. Thus, the minimization criterion for a regularization training algorithm is of the form

$$M = \alpha W + \beta S \tag{2}$$

where α and β are hyper-parameters that determine the relative importance of minimizing W and S, respectively. If $\alpha \ll \beta$ then the training algorithm will develop a network that accurately predicts the training data. If $\alpha \gg \beta$ then the training algorithm will develop a network that has minimally sized weights, thus, producing a smoother network surface at

the expense of larger errors with respect to the training data. Appendix A gives a more thorough explanation of the Bayesian interpretation of the regularization training scheme.

A difficult task in regularization is to determine values of α and β that will yield a network response that accurately fits the training data, and is still reasonably smooth for good generalization on novel inputs. MacKay[7] interpreted the minimization criterion of Eq. 2 under a Bayesian framework to determine optimal values of α and β that produce networks with excellent generalization properties. MacKay[7] showed that optimal regularization parameters could be found using

$$\alpha^{mp} = \frac{\gamma}{2W} \tag{3}$$

$$\beta^{mp} = \frac{N-\gamma}{2S} \tag{4}$$

where γ is interpreted as the effective number of parameters in the current NN, and is calculated

$$\gamma = N_W - \alpha \mathrm{Trace}(\mathbf{A}^{-1}). \tag{5}$$

Here $\mathbf{A}$ is the Hessian matrix of the NN objective function (M), with respect to the parameter weights ($\mathbf{w}$). The training algorithm minimizes $M(\mathbf{w}^{mp}|\alpha,\beta)$ with respect to $\mathbf{w}$, then updates α and β with respect to Eqs. 3 and 4. This iteratively continues until convergence upon hyperparameter values. The general training algorithm here is the same as Foresee and Hagan.[26] When several independently trained NN models have been developed the log evidence is used to choose the single best model. The log evidence is presented and discussed further in Appendix B (Eq. B5), however, it is sufficient to say that the log evidence is simply the Bayesian criterion used to select the single best NN. Appendix B also gives further details and derivations on how this framework is used to optimize the values of α and β.

Training with repeated experiments

It is common in an experimental design that certain experiments are repeated several times. These repeated experiments are used to estimate the noise levels in the observed output, and determine how well hypothetical models fit the data. The Bayesian NN training methods that were first described by MacKay[7] are often capable of accurately estimating such noise levels without repeated experiments; however, data from repeated experiments can be used to improve on the Bayesian training approach. The information from a set of repeated experiments can be easily adapted into such methods by using a prior distribution over β (the inverse variance). Such a prior distribution can be easily formed from the measurements of repeated experiments. For example, assume that the measured response is generated from some true function of the input variables ($f_t(\mathbf{x})$) plus additional noise (ε)

$$y_i = f_t(\mathbf{x}_i) + \varepsilon_i \tag{6}$$

where the error is distributed normally with zero mean and a standard deviation of ($\varepsilon_i \sim \mathcal{N}(0,\sigma_y)$). If N_R experiments are performed under the same conditions ($\mathbf{x}_1 = \mathbf{x}_2 = \cdots = \mathbf{x}_{N_R}$), then the posterior distribution for β is

$$\Pr(\beta|D_{N_R}) \propto \sigma_y^{1-N_R} \exp\left(-\frac{\sum_{i=1}^{N_R}(y_i-\bar{y})^2}{2\sigma_y^2}\right). \tag{7}$$

This posterior density can then be used empirically as a prior density for the training of neural network weight parameters. When this prior information is included into the optimization of β Eq. 4 then becomes

$$\beta^{mp} = \frac{N-\gamma+N_R-1}{2S\left(\mathbf{w}^{mp}\right)+\sum_{i=1}^{N_R}(y_i-\bar{y})^2} \tag{8}$$

where γ is the same as Eq. 5. The optimal estimate for α remains the same as Eq. 3. A more detailed explanation of the derivation of Eq. 8 is shown in Appendix C. Appendix C also presents the new version of the log evidence that is to be used to select a single best NN when prior information on β is used (Eq. C1).

Estimation of error in the network response

Estimating the error in the network response is an important step in formulating the criteria later used to design experiments. Each prediction the network makes will have two sources of uncertainty. One originating from the estimated noise level of the underlying function (estimated via β) and the second originating from the uncertainty in the estimated network weights (estimated via $\mathbf{A}$).[10] For the network prediction of the input $\mathbf{x}_i$ we have

$$P(y_i|D_N,\mathcal{A}) = \mathcal{N}\left(f\left(\mathbf{x}_i|\mathbf{w}^{mp},\mathcal{A}\right),\sigma_{y_i}\right) \tag{9}$$

where the probability of y_i is distributed normally around the network output, and the standard deviation is determined by

$$\sigma_{y_i} = \sqrt{\frac{1}{\beta}+\mathbf{g}_i^T\mathbf{A}^{-1}\mathbf{g}_i} \tag{10}$$

where $\mathbf{g}_i$ is the network sensitivity at $\mathbf{x}_i$ ($\mathbf{g}_i = \partial f(\mathbf{x}_i \mid \mathbf{w}^{mp},\mathcal{A})/\partial\mathbf{w}$). It should be noted that Eq. 9 is only a normal approximation, and does not account for multimodal optima of $\mathbf{w}^{mp}$.

Maximizing information gain

It is desired that collected data estimate model parameters as accurately as possible. In the Bayesian framework, the accuracy of estimated parameters is quantified by the entropy of the posterior distribution ($\Pr(\mathbf{w}|D_N,\mathcal{A})$), where the network weights ($\mathbf{w}$) are the parameters of interest. The entropy of a distribution is a measure of its uncertainty and calculated as

$$H_{D_N} = \int \Pr(\mathbf{w}|D_N,\mathcal{A}) \log\frac{m}{\Pr(\mathbf{w}|D_N,\mathcal{A})} d^{N_W}\mathbf{w} \tag{11}$$

where m is a constant that assures that the quantity in the log is dimensionless and essentially of no concern to the final results.[13] Small values of H_{D_N} indicate that the parameter values are precisely known. It is the underlying task of experimental designs to minimize H_{D_N}.

By choosing different input values to experiment with, we get different amounts of information, which is quantified by the entropy H_{D_N}. Therefore, we wish to choose experiments in a manner that will minimize H_{D_N}. For example, consider the following simplified network structure: $y = wx+\varepsilon$, where there is one input (x), one output (y), one weight parameter (w), and additional Gaussian noise ($\varepsilon_i \sim \mathcal{N}(0,0.1)$). Data must be collected at values of x that minimize the entropy of the posterior distribution of the unknown parameter $\Pr(w|D_N)$. Figure 1a and b show two different data sets collected for this system, where the true parameter value is equal to one (shown with solid line). Data set Da_3 consists of three simulated experiments, with additive noise collected at a value of x that is close to the origin. Db_3 consists of three simulated experiments with identical additive noise collected at a value of x, that is further from the origin. Notice that there is a wider range of possible lines that pass through Da_3 (shown with dashed lines). Thus, there is a larger degree of uncertainty in the true value of the parameter value (w). This degree of uncertainty can be seen in the posterior distributions of w shown in Figure 1 c and d. A wider distribution indicates we are less certain of the true value of w, and in return results in a larger entropy value. For this example Hb_3 is less than Ha_3. Placing experiments further from the origin results in more informative data, because we will learn more about the true value of w. Similarly, experiments can be placed in the input space of NN models that will yield greater information, however, such placements will not be as straight forward if the NN response is nonlinear.

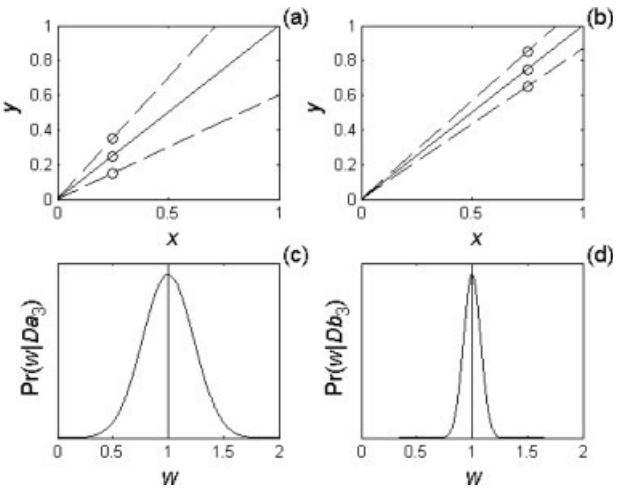

Figure 1. The fundamental properties of a data set are shown that result in optimal parameter estimation of a model.

Different data sets are compared for a problem of linear regression through the origin. (a) shows data that has been collected close to the origin (Da_3). It can be seen that a wide range of slopes are possible given the data, (b) shows data that has been collected further from the origin (Db_3). A much smaller range of slopes is possible given this different data set. (c) and (d) show the Bayesian posteriors for the slope (w) given the different data sets (Da_3 and Db_3, respectively). Db_3 more precisely predicts the true slope of the system ($w = 1$). Thus data further from the origin will more efficiently collect data. The entropy from a Bayesian posterior can quantify this property. $\Pr(w|Db_3)$ has a smaller entropy than $\Pr(w|Da_3)$.

If some data (D_N), has been collected to train a NN, then the entropy of the weights in that NN can be estimated using a Taylor series expansion around the optimum (Eq. B3) to solve for the integral in Eq. 11

$$H_{DN} = \frac{N_W}{2}(1 + \log 2\pi) + \frac{1}{2}\log(m^2 \det \mathbf{A}_N^{-1}). \tag{12}$$

Next we wish to collect an additional data point that will minimize the resulting new entropy (H_{N+1}), or equivalently maximize the change in entropy ($\Delta H = H_N - H_{N+1}$). This will optimally reduce the uncertainty in the estimated NN weight parameters after they have been trained on the $N + 1$ data points. The Hessian of this newly trained network can be approximated as

$$\mathbf{A}_{N+1} \cong \mathbf{A}_N + \frac{1}{\sigma_y}\mathbf{g}\mathbf{g}^T \tag{13}$$

where $\mathbf{g}$ is the sensitivity to the network output at the newly collected data point. It follows that the estimated change in entropy after this additional data point is collected is

$$\Delta H = \frac{1}{2}\log(m^2 \det \mathbf{A}_N^{-1}) - \frac{1}{2}\log(m^2 \det \mathbf{A}_{N+1}^{-1}). \tag{14}$$

MacKay[13] showed that Eq. 14 reduces to what is referred to as the total information gain (TIG)

$$\text{TIG} = \frac{1}{2}\log(1 + 1/\sigma_y \mathbf{g}^T \mathbf{A}_N^{-1}\mathbf{g}). \tag{15}$$

Notice that the information gained from a single new experiment increases monotonically with the size of the error bars shown in Eq. 10, thus, we learn the most about the NN weights in regions of space where we are most uncertain about the value of the output.

To generalize this result to a batch of new experiments we consider how the entropy will change after a series of B new experiments. An estimation of the total information gain after a batch of experiments is then

$$\begin{aligned}\text{BTIG} &= \sum_{b=1}^{B} H_{N+b-1} - H_{N+b} \\ &= \sum_{b=1}^{B} \frac{1}{2}\log\left(1 + 1/\sigma_y \mathbf{g}_b^T \mathbf{A}_{N+b-1}^{-1}\mathbf{g}_b\right)\end{aligned} \tag{16}$$

where $\mathbf{g}_b$ is the network sensitivity of the b^{th} new data point, and $\mathbf{A}_{N+b-1}$ is the Hessian immediately before the b^{th} new datum has been observed. Each of these Hessian matrices can be approximated by iteratively updating $\mathbf{A}$ according to the sensitivities

$$\begin{aligned}\mathbf{A}_{N+1} &\cong \mathbf{A}_N + \frac{1}{\sigma_y}\mathbf{g}_1\mathbf{g}_1^T \\ \mathbf{A}_{N+2} &\cong \mathbf{A}_{N+1} + \frac{1}{\sigma_y}\mathbf{g}_2\mathbf{g}_2^T \\ &\vdots \\ \mathbf{A}_{N+B} &\cong \mathbf{A}_{N+B-1} + \frac{1}{\sigma_y}\mathbf{g}_B\mathbf{g}_B^T\end{aligned} \tag{17}$$

The batch-total information gain (BTIG) can then be used as a criterion to select any number of new experiments to perform.

Maximizing relative information gain

We have so far reviewed the TIG first put forth by MacKay[13] and have generalized this criterion for a batch of experiments. Optimizing either one of these metrics attempts to maximize the information gained about the entire output range of the network, however, we are primarily concerned with maximizing the information gained with respect to new optima of the underlying function. To do this we propose using the relative information gain (RIG), which is a simple extension of the TIG concept,

$$\mathrm{RIG} = \exp\left(-\frac{(y_{opt}-y)^2}{\sigma_{opt}^2}\right)\frac{1}{2}\log\left(1+\frac{1}{\sigma_f^2}\mathbf{g}^T\mathbf{A}^{-1}\mathbf{g}\right). \quad (18)$$

Here the TIG has been multiplied by a factor that penalizes output values that are not close to the current network optimum (y_{opt}). The scaling factor σ_{opt}^2 determines how severely to penalize such outputs. A very low-value will strongly stress choosing new experiments to run that are currently close to the network optimum. A larger value of σ_{opt}^2 will allow for riskier experimentation to occur. We suggest setting σ_{opt}^2 proportional to the variance of the observed responses. For example, setting σ_{opt}^2 equal to

$$\sigma_{opt}^2 = 4\frac{\sum_{i=1}^{N}(y_i-\bar{y})^2}{N-1} \quad (19)$$

essentially specifies a prior preference that states that we are 68% sure that we do not want to try any experiments that are greater than the current observed experimental mean ($\bar{y}$) (note: this statement assumes minimization of the NN output is the objective).

Generalizing the RIG to a batch of experiments can be done by multiplying each incremental information gain by its corresponding penalization factor

$$\mathrm{BRIG} = \sum_{b=1}^{B}\exp\left(-\frac{(y_{opt}-y_b)^2}{\sigma_{opt}^2}\right) \times\frac{1}{2}\log\left(1+1/\sigma_f\mathbf{g}_b^T\mathbf{A}_{N+b-1}^{-1}\mathbf{g}_b\right) \quad (20)$$

where y_b is the network output, and $\mathbf{g}_b$ is the network sensitivity to the b^{th} new data point.

Results

Three case studies are used to illustrate the utility of the presented material. Case study one uses a simple 1-D regression problem to demonstrate the three important contributions of this work: the use of informative priors for noise hyperparameters in Bayesian regularization, the use of the RIG compared to the TIG, and the use of the BRIG to conduct sequential experimental designs. Case study two uses the BRIG to conduct a sequential experimental design for a more complex 2-D optimization problem. This approach is shown to outperform a sequential design using the TIG, and a traditional five level full factorial design. Case study three validates the use of informative priors for noise hyperparameters and the performance of the RIG over the TIG using a real historical fermentation database.

Case Study One: 1-D regression

In this first case study we implement the algorithms presented ealier on an arbitrary 1-D regression problem. The generating function of the data is

$$y = \exp(2x) + 2\exp\left(-10(x-0.1)^2\right) - 6\exp\left(-25(x-0.3)^2\right) + \mathcal{N}(0,0.25) \quad (21)$$

where there is one independent variable defined between negative one and one ($-1 \le x \le 1$) and additive Gaussian noise with a mean of zero and standard deviation of 0.25. Figure 2 shows this underlying function with the upper and lower 95% confidence intervals along with a single set of generated data. These data (D_7) contains seven simulated data points (x = [−1,−0.5, 0, 0, 0, 0.5, 1] y = [−0.4048, 0.3813, 1.9717, 2.2425, 2.4064, 1.3167, 7.1249]). Next we would like to use these data to approximate the underlying function.

To show the advantage of using an informative prior over β, two different NN models were trained using D_7. Both models contained one hidden layer with ten tansig nodes and one linear output node. The first network (NET_1) was trained according to Foresee and Hagan,[26] using Eq. 3 and Eq. 4 to update the hyperparameters; 50 networks were trained using different initialized weights, and the network with the maximum log evidence (calculated from Eq B5) that included a uniform prior over β was chosen. The second network (NET_2) was trained in an identical manner, except Eq. 8 was used to update β (instead of Eq. 4), and the log evidence that included an informative prior over β was used to select the final network structure, Eq. C1). Using Eq. C1) instead of Eq. B5 utilizes information about the experimental noise

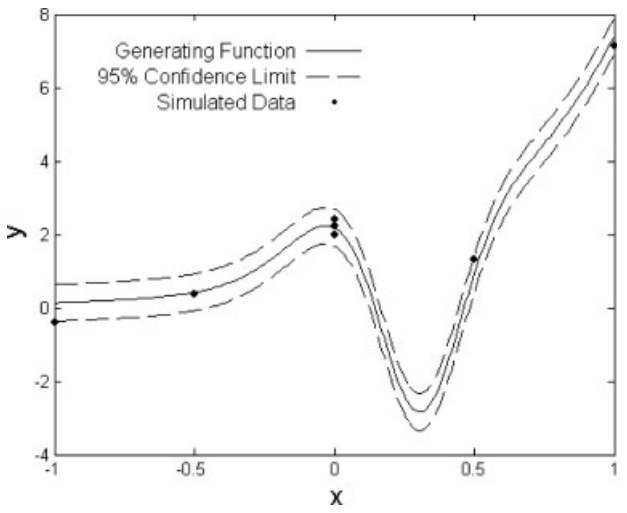

Figure 2. An arbitrary function with one independent (*x*) and one dependent (*y*) variable.

The underlying function has the form $y = \exp(2x) + 2\exp(-10(x-0.1)^2) - 6\exp(-25(x-0.3)^2)$, where Gaussian noise is added to simulated data points. A single simulated data set is also shown that is used to train an initial neural network model.

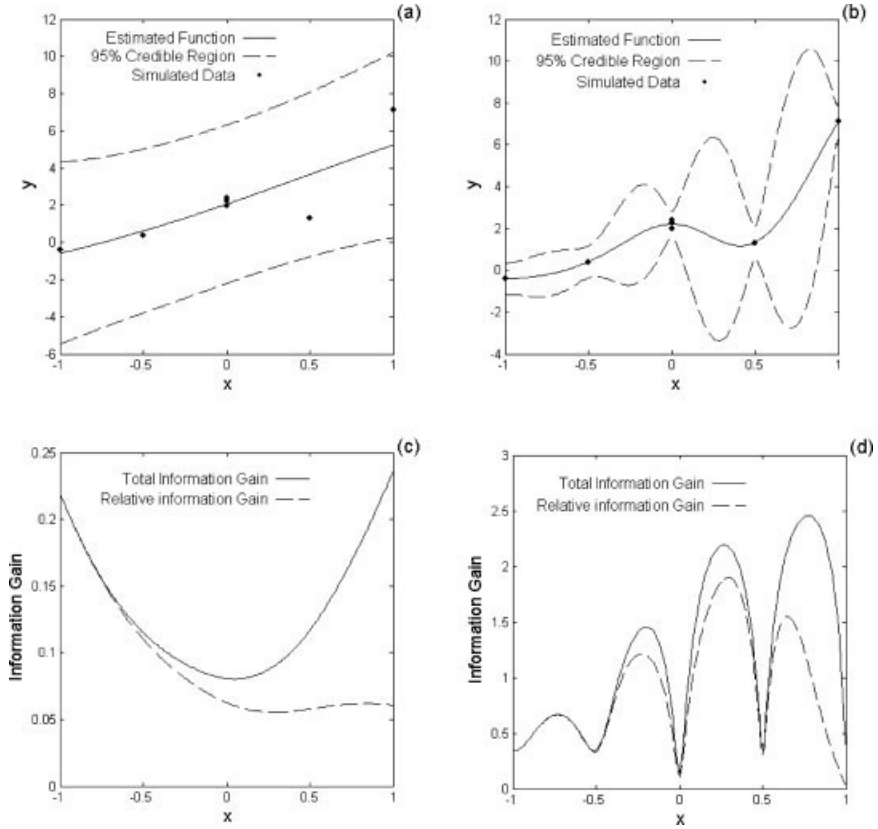

Figure 3. Results from two different neural networks trained on the same data set of case study one.

(a) and (c) correspond to NET_1, where an uninformative prior over the noise parameter β is used. (b) and (d) correspond to NET_2, where an informative prior over the noise parameter β is used. NET_1 is essentially a linear model. NET_2 does not predict the underlying function exactly, but is able to determine that there are some nonlinear relationships occurring. For both networks the total information gain monotonically increases with respect to the 95% credible regions, and the relative information gain is a function of both the total information gain, and the predicted output value of the network.

level obtained before training a NN. An informative prior was empirically formulated using the repeated simulations

$$P(\beta|D_{N_R}) \propto \beta \exp\left(-\frac{\beta}{2}0.0964\right) \tag{22}$$

where D_{N_R} are the three data points simulated at $x = 0$, and the sum of squared total error $(\sum_{i=1}^{N_R}(y_i - \bar{y})^2)$ of those three outputs is 0.0964.

Figure 3a and b show the output response, and 95% credible regions for NET_1 and NET_2, respectively. NET_1 is essentially linear, while NET_2 is slightly more complex, but fits the presented data more accurately. Both models are reasonable given the amount of data that has been collected, however, NET_1 estimates σ_y to be 1.967, while NET_2 estimates σ_y to be 0.270. Using the three repeated experiments at $x = 0$, we find that the best guess for σ_y is 0.220. Given such information from repeated experiments we qualitatively may prefer models that predict the noise level to be closer to this value. The log evidence of Eq. C1 takes this information into account and selects the more reasonable model. We can also see that NET_2 is a closer approximation to the true generating function shown in Figure 2, thus, validating the use of an informative prior in this case.

Next we would like to use the developed models to suggest a single new point that will hopefully reveal a new optimum of the function. To do this we calculate the TIG and RIG for each network, which are shown in Figure 3c and d. Notice the TIG monotonically increases with the size of the credible regions of its respective model, while the RIG is a function of both the size of the credible regions and the value of the network response. The RIG is scaled down as the network response increases (when minimization is the objective). The TIG of NET_1 suggests that the optimal experiments should be performed at the edges of the input space (negative one and one). This is the same result one would reach for a traditional set of optimally designed experiments assuming linearity. The TIG of NET_2 suggests there are several different places in the interpolation space, where optimal experiments can be performed. Such places lie in the sparsest regions of collected data, however, the TIG is not just a measure of the distance from collected data; it also increases as the complexity of the network response increases. The global optimum of the TIG for NET_2 is at, $x = 0.77$, this is because that region of input space is sparse, and the network response in that region is rapidly changing (hence, more complex). However, the output values are also relatively high in this region, therefore, the RIG scales this region down. The global optimum of the RIG is

at $x = 0.31$. This region of space is a compromise between being highly informative (high TIG), and having a low-network response. Notice that $x = 0.31$ is very close to the true optimum of the underlying function shown in Figure 2. Also note that both NET_1 and NET_2 predict the global optimum to be at $x = -1$. Thus, using the RIG will yield the quickest path to optimization.

Instead of performing just a single new experiment, suppose that we wish to perform batches of experiments until we have successfully determined the global optimum. To do this we will use the BRIG in conjunction with NET_2, which has the form

$$\text{BRIG} = \sum_{b=1}^{B} \exp\left(-\frac{\left(y_{opt} - NET_2(x_b)\right)^2}{\sigma_{opt}^2}\right) \times \frac{1}{2}\log\left(1 + \beta_2 \mathbf{g}_b^T \mathbf{A}_{N+b-1}^{-1} \mathbf{g}_b\right). \quad (23)$$

However, we are unsure how many new experiments to perform at a time (B). Table 2 shows experimental designs for various values of B along with their corresponding BRIG values. The size of B will typically be constrained by the physical capabilities of the available equipment (for example, a company may only own four vessels with which to perform experiments), however, a simple table like Table 2 may help in deciding how many experiments to perform at one time. Notice that the BRIG monotonically increases at a decreasing rate with respect to B, with the largest jump in information gain at $B = 1$ (0 to 2.0974). This suggests that performing one new experiment at a time might be the most informative; this is typically true for most systems, however, in a process development environment it is most efficient not to let available equipment become idle. Note that when $B = 2$ the experimental design essentially includes the two highest peaks in the RIG, however, when $B = 3$ the experimental design includes points that become centered about the optimum of the RIG; the BRIG does not just find all the local optima of the RIG. For example, when $B = 3$ two points get centered around the global optima ($x = 0.217$ and $x = 0.369$), and one point gets placed near the other local optima ($x = 0.653$). While the optimal value of B is typically 1 (one experiment at a time), the calculations shown in Table 2 can be helpful in determining the best B value for a given situation.

Table 2. Chosen Experimental Designs for Different Batch Sizes (B)

Batch Size (B)	Input Values (x)	BRIG
1	0.310	2.0974
2	0.304, 0.656	2.671
3	0.217, 0.369, 0.653	3.0883
4	−0.298, 0.231, 0.370, 0.651	3.4286
5	−1.000, −0.300, 0.231, 0.370, 0.651	3.7685
6	−1.000, −0.660, −0.266, 0.226, 0.364, 0.648	4.0652

Each batch is designed using the BRIG (Eq. 23), when NET_2 is trained using the original seven data points shown in Figure 2.

Case Study Two: 2-D modified Himmelblau function

In this second case study we validate the use of the BRIG to perform sequential designs on a more complex 2-D optimization problem. We show that the BRIG outperforms a sequential design based on the BTIG, and a traditional five level full factorial design. The function under consideration is the modified Himmelblau function[18]

$$y = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2 + x_1 + 3x_2 + 57 + \mathcal{N}(0,2) \quad (24)$$

where $-5 \le x_1 \le 5$ and $-5 \le x_2 \le 5$. Figure 4a and b are surface and contour plots of this function. Figure 4b marks the three local optima (output values of 65.9, 54.9, 63.5), along with a single global optimum (43.3) for this function. The general goal in this example is to determine the global optimum with as few experiments as possible.

First consider a traditional approach of using a five-level full factorial design with three repeated experiments at the center point. This design was simulated using Eq. 24, and then used to train a NN, where the informative prior over β was formulated using the three repeated experiments in the center, and Eq. B5 was used to select the final NN weights. Figure 4c shows the experimental design superimposed over the contours of the trained network. To test the overall predictive capabilities of the NN a test set was generated ($\mathbf{x}_{\text{test}}$) using a 101 level full factorial design, where the corresponding outputs were calculated from Eq. 24 without additional Gaussian noise. The R^2 value of this NN with respect to this large test set is 0.86. The general shape of the modified Himmelblau is captured, however, the global optimum of the NN does not match the actual global optimum (Figure 4c).

Now consider a sequential design starting from a three level factorial design (with three repeated experiments at the center point) and sequentially building up to 27 experiments in batches of four. This will result in the same number of data points in the final training set of the factorial design of Figure 4c. Figure 4d shows the starting design superimposed over the resulting NN model that was trained in an identical manner as was used for the factorial design data. The network surface does not closely resemble the true modified Himmelblau function, however, it does contain some useful information that can be used to design the next set of experiments. For example, we have learned that the corners of the underlying function are much greater than the center point. Figure 4d also shows the next set of experiments suggested by optimizing the BTIG ($\diamond$), and the BRIG ($\times$) for four experiments ($B = 4$). Notice that the BTIG suggests more points that are on the edge of the input space while the BRIG suggests points that are relatively well centered about the current models global minimum. Continuing to collect data according to the BTIG (and retraining the NN after each batch) results in the model surface shown in Figure 4e, and has a R^2 value with respect to the test set of 0.97. Continuing to collect data according to the BRIG results in the model surface shown in Figure 4f, and has a R^2 value with respect to the test set of 0.99. Both final experimental designs are superimposed over the model surfaces. Notice that the data collected using the BTIG has not identi-

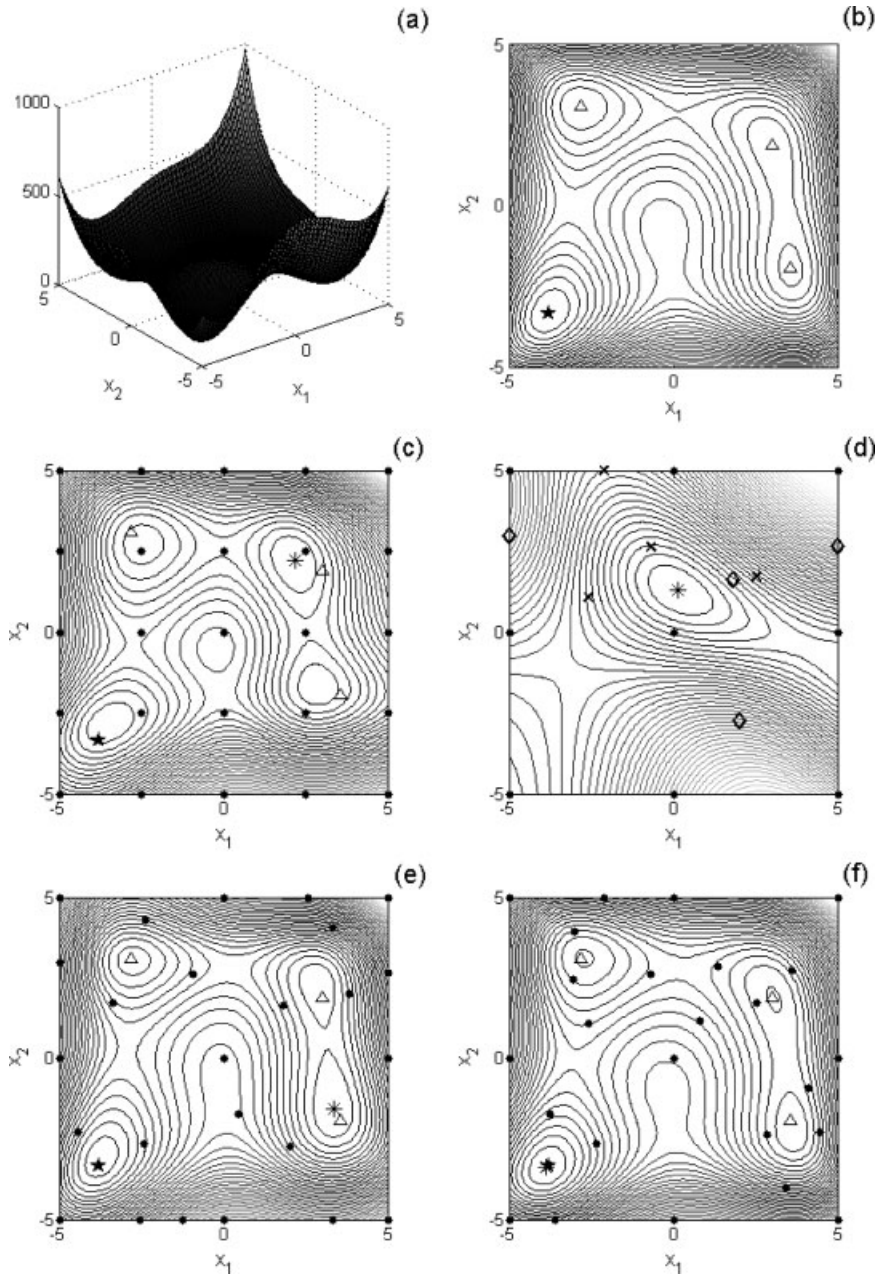

Figure 4. Results from case study two.

(a) and (b) show a surface and contour plot of the underlying function (modified Himmelblau) along with the local (△) and global (★) optimal; (c) shows a NN model developed from a training set that was collected according to a five-level full factorial design (●) with three repeated experiments at the center point for a total of 27 data points. The global optimum of the NN (⁎) is not near the true global optimum (★); (d) shows a NN model developed from a training set that was collected according to a three-level full factorial design (●) with three repeated experiments at the center point. Further experiments to be performed according to the BTIG (◇) along with the BRIG (×) are also shown; (e) shows the resulting NN from a sequential experimental design (●) of four batches using the BTIG (four experiments in each batch). (f) shows the resulting NN from a sequential experimental design (●) of four batches using the BRIG (four experiments in each batch). Both (e) and (f) use 11 points (shown in (d)) as the starting training set for a total of 27 data points. Only the sequential design using the BRIG, and (f) was able to correctly identify the global optimum.

fied the true global minimum, although, the network surface is more accurate than the surface modeled with the five-level factorial design. The resulting surface from the BRIG has identified the true minimum of the modified Himmelblau. Overall this example shows that sequential designs using the BTIG or the BRIG are significant improvements over a traditional factorial design. Furthermore, the sequential design using the BRIG revealed the quickest path to true global optimum.

Case Study Three: Optimization of fermentation protein yield

In this third example we analyze data from an experimental recombinant *E. coli* fermentation database generated in

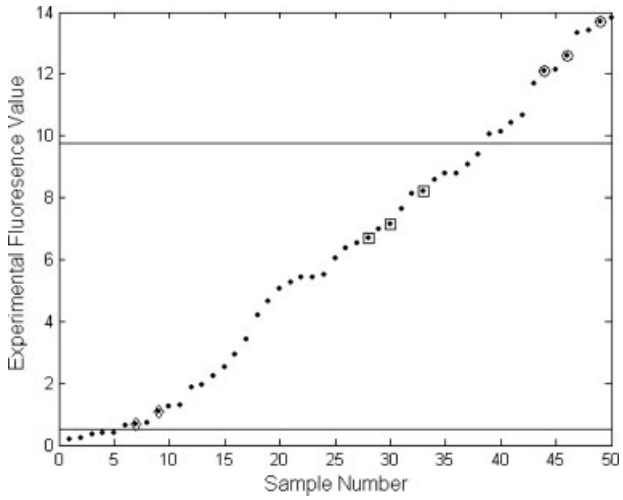

Figure 5. Experimental fluorescence value of 50 fermentations listed from smallest to greatest experimental fluorescence value.

The open diamonds, squares, and circles mark the output values that were attained from identical fermentation conditions. The horizontal lines partition the total data set into training and testing sets. The data points that are between the horizontal lines are used to make a training set and the points that are outside are used to make a testing set.

our laboratory. In this fermentation, *E. coli* produces green fluorescent protein (GFP). Fermentations were performed under various operating conditions (for example, temperature, pH, media concentrations). The goal of this example is to build a model that predicts the protein yield of novel combinations of operating conditions given a historical database. Once this model is developed it is desired to use this model to suggest new operating conditions that will potentially optimize the system.

Fifty fermentations were performed using a five-level partial factorial design of 14 variables. From these 50 fermentations seven out of the 15 input variables were determined to be important using data mining techniques.[24,25,27] Only these seven input variables were used here to model the output in order to simplify the results. Several experimental conditions were repeated multiple times to assure that the noise level did not overwhelm the signal. Figure 5 plots the observed fluorescence level of each of the 50 fermentations. The repeated experiments are marked with their respective symbols to show that the variability of the entire database is greater than the variability of a single experimental condition.

Next we separated the top twelve and lower five fermentation conditions into a testing set. Three of the top twelve are from repeated experimental conditions, these three outputs were averaged, thus, reducing the test set by two data points (total of 15 test points). The fluorescence levels used to partition the data are also shown in Figure 5, thus, only data points between the solid lines in Figure 5 were included into the training data. Next we developed a NN model using the training data, and a prior over β that was formed using the repeated experiments of the training data set. Figure 6a shows the experimental values of the test and training sets compared to the network predictions. The training data is modeled very accurately, while there are large errors with respect to the testing set. However, the model is able to distinguish between high-and low-test points. Notice that the maximum NN prediction (arrowed point) corresponds to the maximum of the training set, and none of the test points are predicted to be higher than this point, even though they are higher in reality. Thus, the NN surface does not suggest any novel optima even though they may exist (and in this case they do).

While the NN surface was not able to suggest any new optima the TIG can be used potentially to suggest new experiments to run that will improve our understanding of the NN surface. Figure 6c shows the experimental values of the test and training sets with respect to the TIG. Notice that all of the test points have higher TIG values than all of the training points. This is because the TIG recognizes that the test set points are in regions of space that have not yet been explored, thus, more informative. However, the highest TIG value (arrowed point) belongs to a test set point that has a low-experimental fluorescence value. Thus, maximizing the TIG, in this case, suggests experimentation on conditions that yield low-fluorescence values. Figure 6e shows the experimental values of the test and training sets with respect to the RIG. Notice that the point with the maximum RIG value (arrowed point) corresponds to the maximum fluorescence value of the test set. Thus, maximizing the RIG suggests novel input conditions that result in higher fluorescence values than those that were observed in the training set. This type of result illustrates how RIG can be used for more efficient and cost-effective process development.

The results shown in Figure 6a, c, and e are dependent on the use of an informative prior over β. Figure 6b, d, and f show analogous results when an uninformative prior is used. Figure 6b, d, and f use Eq. B5 to select the final network structure, while Figure 6a, c, and e use Eq. C1. Notice that there are larger errors in the NN predictions (Figure 6b) with respect to the training set. This is because the NN is overgeneralizing and predicting the experimental system to have higher noise levels. Also notice that maximizing any of the criteria (NN surface, TIG, or RIG) does not result in suggesting any of the points in the high-test set.

Overall, this example shows the importance of using the RIG and an informative prior over β. Out of the three criteria considered for choosing new experiments (NN surface, TIG, RIG) only the RIG (when combined with an informative prior over β) resulted in suggesting a point from the test set that resulted in an optimal fluorescence value.

Discussion

The aim of this work has been to develop strategies that can design experiments for nonlinear systems to reveal new optima. We introduced several modifications to the criteria presented by MacKay[13] that help identify new optima of nonlinear systems. We first generalized the total information gain (TIG) to quantify the information gained from a batch of new experiments; this is quantified using the batch information gain (BTIG). We then modified the TIG with respect to the neural network output, which penalizes the TIG and BTIG for suggesting experiments that have a low-potential for finding new optima, this is quantified in the relative infor-

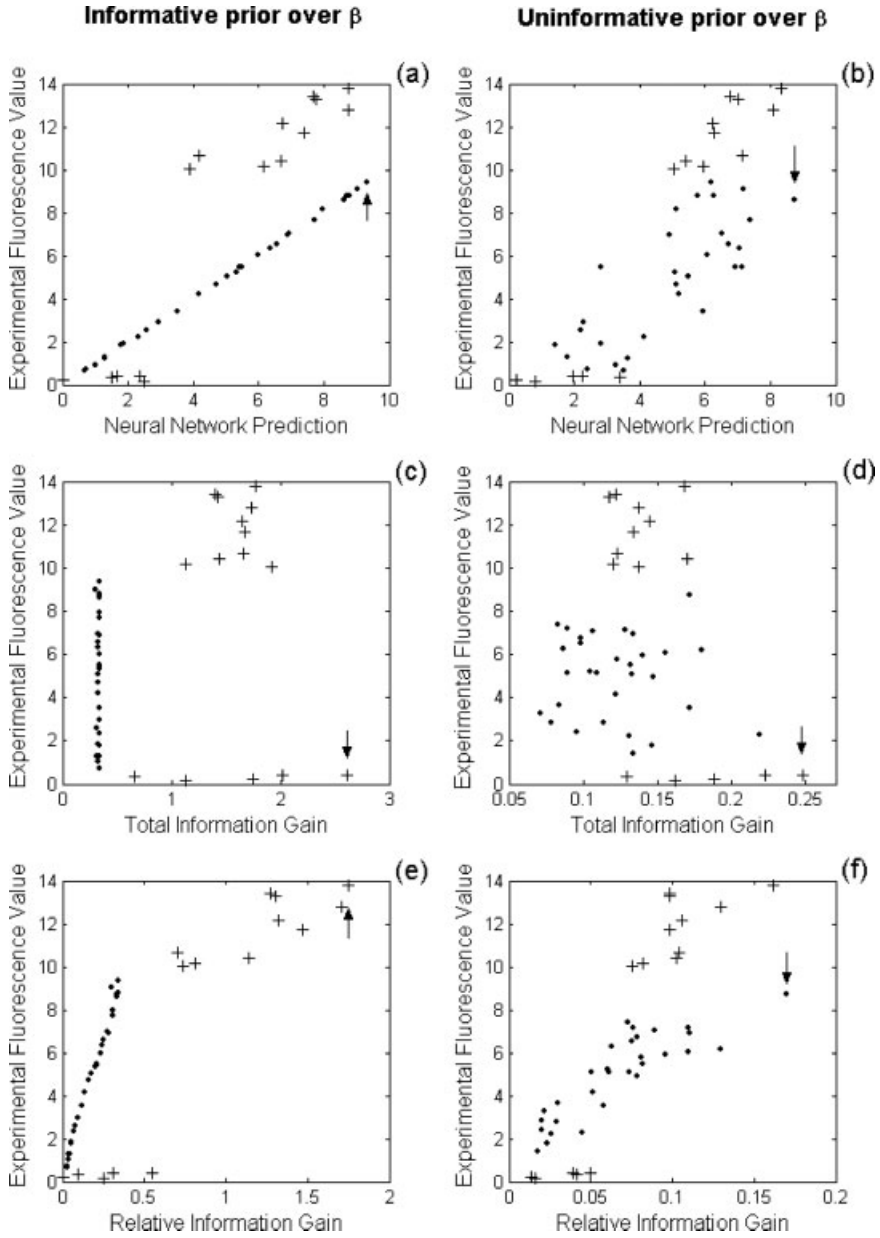

Figure 6. Results from developing NNs using the training data and predicting the test data shown in Figure 5.

(a), (c), and (e) show the experimental fluorescence values vs. three different criteria (NN surface, TIG, RIG) for a NN developed using an informative prior over β. (b), (d), and (f) all correspond to a NN developed using an uninformative prior over β. Arrowed points indicate predicted optima according to each criterion. Notice that the RIG associated with the NN using the informative prior is the only criterion that is able to predict points in the test set that will increase the experimental fluorescence value.

mation gain (RIG), and the batch-relative information gain (BRIG). We showed in three case studies that these criteria were able to be effectively used to find optima of nonlinear systems. However, the results of this work are also dependent on the use of repeated experiments to form an informative prior over the noise hyperparameter included in Bayesian regularization training methods.

An important analogy should be stressed between the first and third case studies. In both cases a NN was trained using the available data, and an informative prior over β. When such a network was trained the test data was accurately predicted, however, there were very large error bars in regions of sparse data. Both of these networks had optimal output values that were equivalent to input sets that had already been collected. When new experiments were suggested by optimizing the RIG, then new optima were found. On the contrary, when the TIG was optimized a new optima to the output was not found. It is difficult to determine what is exactly causing such phenomena in the 7-D problem of case study three, however, in the 1-D problem of case study one, it can be seen that the nonlinear nature of the interpolation space has a large degree of uncertainty. When networks were trained for both data sets that did not include informative priors over the noise parameters, over generalization occurs

(Fig 3a and Figure 6b). For such a scenario, the uncertainty of the NN weights is seen to be due to the high-noise levels. Thus, the model will suggest repeating experiments on the edges of the input space to more accurately estimate the true output values in these regions. However, precisely knowing the output values on the edges will only result in optimal predictions of the interpolation space if the interpolation space is linear.

Case study two exemplifies the virtues of using the BRIG for sequential experimental design for nonlinear systems. The BRIG was able to select new experiments to perform that were roughly centered around the local optima, thus, improving the prediction of the global optimum. The BTIG performed rather well, however, still failed in determining the global optimum. An important point to be made in this case study is that evenly spacing out experiments (as is done in factorial designs) will not necessarily yield an optimal amount of information for nonlinear systems. A sequential design can drastically improve the modeling capabilities of the developed model. Furthermore, such designs can be accomplished using any of the presented criteria (TIG, RIG, BTIG, or BRIG). The use of any of these criteria offers several advantages over traditional sequential designs. First, any amount of arbitrary data can be used. Typically response surface methodology or sequential factorial design procedures use only a subset of the available data at each step in the design. This is because of the hill climbing (or gradient based) nature of these algorithms. The approaches described here can utilize databases of any size or of arbitrary conformations as might be commonly found in industrial manufacturing or process development databases. Furthermore, these criteria do not rely on gradient methods, and are capable of designing experiments in multiple directions with respect to the placement of the current optima. This results in more robust designs that may help to avoid getting stuck in local optima.

Another advantage to these criteria is that they offer a single objective function that can be used to design experiments, thus, eliminating the need to use multiple design strategies. For example, Cockshott and Sullivan[28] successfully optimize the medium of *Aspergillus nidulans* fermentations using an array of traditional experimental design tools, including Plackett-Burman and factorial designs along with response surfaces and ridge analysis. Each of these techniques are required to handle various situations. For instance, ridge regression is used to explore response surfaces when they do not result in unique optima. However, such a technique is not required for the presented criteria.

We have in general shown that the presented criteria are efficient at collecting new data to find new optima. However, there are several drawbacks to the presented method. First, it cannot be used to suggest how to begin collecting data. Traditional methods of experimental designs should prove to be sufficient; however, if inefficient starting experiments are performed, the presented methods may be rendered useless until sufficient data is acquired. Another major drawback is that if a poor network structure is being used then that model will suggest poorly designed experiments. This is emphasized in Figures 3a and Figure 6b of case studies one and three, respectively. In both cases over generalization occurs which prevents the model surface, TIG, or RIG from successfully suggesting new optima. However, one cannot guarantee that over generalization or over fitting will not occur. Despite the possible drawbacks to the presented material, we have shown that these criteria can efficiently identify new optima, whether new experiments are performed one at a time, or several simultaneously.

Acknowledgments

The authors would like to thank the California Dairy Research Foundation, University of California Discovery Grant Program, and the Viticulture Consortium for the financial support of this research.

Notation

NN = neural network
TIG = total information gain
$BTIG$ = batch-total information gain
RIG = relative information gain
$BRIG$ = batch relative information gain
D_N = observed data with N data points
$\boldsymbol{x}_i$ = input vector of i^{th} observation
y_i = output of i^{th} observation
S = sum of squared error
f = NN function
$\boldsymbol{w}$ = NN parameter vector
$\mathcal{A}$ = NN architecture
W = sum of squared weights
N_W = total number of weights
M = regularization training criterion
α = regularization hyper-parameter for W
β = regularization hyper-parameter for S
σ_W^2 = prior variance of NN weights
S_σ = weighted least-squares minimization criterion
σ_i^2 = variance for the i^{th} observed output value
σ_y^2 = variance for all observed output values
$\boldsymbol{A}$ = hessian matrix of M
$\boldsymbol{w}^{mp}$ = NN weights that minimize M
α^{mp} = value of α that optimizes the posterior assuming $\boldsymbol{w}^{mp}$
β^{mp} = value of β that optimizes the posterior assuming $\boldsymbol{w}^{mp}$
f_t = true function that generates data observations
ε_i = error associated with i^{th} observation
N_R = number of repeated experiments with identical input conditions
$\boldsymbol{g_i}$ = NN sensitivity with respect to weights parameters at $\mathbf{x}^i$
B = number of experiments performed in a batch of experiments
$\mathcal{N}(0,1)$ = normal distribution with mean of zero and unit standard deviation

Literature Cited

1. Fedorov VV. *Theory of optimal experiments*. New York: Academic Press; 1972.
2. Atkinson AC, Donev AN. *Optimum experimental designs*. Oxford New York: Clarendon Press; Oxford University Press; 1992.
3. Hicks CR, Turner KV. *Fundamental concepts in the design of experiments*. 5th ed. New York: Oxford University Press; 1999.
4. Myers RH, Montgomery DC. *Response surface methodology: process and product optimization using designed experiments*. New York: Wiley Inc; 1995.
5. Box GEP, Draper NR. *Empirical model-building and response surfaces*. New York: Wiley Inc; 1987.
6. Mackay DJC. Bayesian interpolation. *Neural Comp*. 1992;4(3):415–447.
7. Mackay DJC. A Practical bayesian framework for backpropagation networks. *Neural Comp*. 1992;4(3):448–472.
8. MacKay DJC. *Information theory, inference, and learning algorithms*. Cambridge, U.K.; New York: Cambridge University Press; 2003.
9. Penny WD, Roberts SJ. Bayesian neural networks for classification: how useful is the evidence framework? *Neural Networks*. 1999;12(6):877–892.

10. Bishop C. *Neural networks for pattern recognition*. New York: Oxford University Press; 1995.
11. Chaloner K, Verdinelli I. Bayesian experimental design: A review. *Statistical Sci*. 1995;10(3):273–304.
12. Loredo TJ, Chernoff DF. Bayesian Adaptive Exploration. In: Feigelson ED, Babu GJ, eds. *Statistical Challenges in Astronomy*. Vol 57. New York: Springer; 2003.
13. Mackay DJC. Information-based objective functions for active data selection. *Neural Comp*. 1992;4(4):590–604.
14. Cohn DA. Neural network exploration using optimal experiment design. *Neural Networks*. 1996;9(6):1071–1083.
15. Cohn DA, Ghahramani Z, Jordan MI. Active learning with statistical models. *J of Artificial Intelligence Res*. 1996;4:129–145.
16. Raju GK, Cooney CL. Active learning from process data. *AIChE J*. 1998;44(10):2199–2211.
17. Lin JS, Jang SS. Nonlinear dynamic artificial neural network modeling using an information theory based experimental design approach. *Ind & Eng Chemistry Res*. 1998;37(9):3640–3651.
18. Chen JH, Wong DSH, Jang SS, Yang SL. Product and process development using artificial neural-network model and information analysis. *AIChE J*. 1998;44(4):876–887.
19. Gelman A, Carlin JB, Stern HS, Rubin DB. *Bayesian data analysis*. 2nd ed. Boca Raton, Fla.: Chapman & Hall/CRC; 2004.
20. Coleman MC, Block DE. Bayesian parameter estimation with informative priors for nonlinear systems. *Aiche J*. 2006;52(2):651–667.
21. Albano CR, Randers-Eichhorn L, Bentley WE, Rao G. Green fluorescent protein as a real time quantitative reporter of heterologous protein production. *Biotechnol Progr*. 1998;14(2):351–354.
22. DeLisa MP, Li JC, Rao G, Weigand WA, Bentley WE. Monitoring GFP-operon fusion protein expression during high cell density cultivation of Escherichia coli using an on-line optical sensor. *Biotechnol and Bioeng*. \ 1999;65(1):54–64.
23. Coleman MC, Block DE. Retrospective optimization of time-dependent fermentation control strategies using time-independent historical data. *Biotechnol and Bioeng*. 2006;95(3):412–423.
24. Coleman MC, Buck KKS, Block DE. An integrated approach to optimization of Escherichia coli fermentations using historical data. *Biotechnol and Bioeng*. 2003;84(3):274–285.
25. Buck KKS, Subramanian V, Block DE. Identification of critical batch operating parameters in fed-batch recombinant E-Coli Fermentations using decision tree analysis. *Biotechnol Prog*. 2002;18(6):1366–1376.
26. Foresee R, Reeves CM. Gauss-Newton approximation to Bayesian regularization. *Proceedings of the 1997 International Joint Conference on Neural Networks*. 1997:1930–1935.
27. Subramanian V, Buck KKS, Block DE. Use of decision tree analysis for determination of critical enological and viticultural processing parameters in historical databases. *A J of Enology and Viticulture*. 2001;52(3):175–184.
28. Cockshott AR, Sullivan GR. Improving the fermentation medium for Echinocandin B production. Part I: Sequential statistical experimental design. *Process Biochem*. 2001;36(7):647–660.
29. Neal RM. *Bayesian learning for neural networks*. New York: Springer; 1996.

Appendix A: Bayesian Framework of NN Regularization

In this framework α is a measure of our prior belief as to complexity of the neural network surface. A small α value indicates a complex surface is expected, and, thus, a large range of NN weights will be needed. A large α value indicates a simple surface is expected, thus, the NN weights should be constrained to have a smaller range. This hyperparameter can then be interpreted as the inverse of the variance to a prior probability distribution that describes the feasible range for the NN weights. If this prior distribution is Gaussian then the resulting prior has the form

$$\Pr(\mathbf{w}|\alpha,\mathcal{A}) = \frac{\exp(-\alpha W)}{(\pi/\alpha)^{N_W/2}} \tag{A1}$$

where the variance of the expected network weights is $\sigma_W^2 = 1/\alpha$, and there are N_W weights in the network architecture $\mathcal{A}$.

The parameter β can be interpreted as the noise level that is present in the observed data. For example, a simple weighted least-squares minimization criterion has the form

$$S_\sigma = \sum_{i=1}^{N} \frac{1}{2\sigma_i^2}\left(f(\mathbf{x}_i|\mathbf{w},\mathcal{A}) - y_i\right)^2 \tag{A2}$$

where σ_i^2 is the estimated noise level for the ith observed output value. It can be seen that the importance of fitting a particular output value decreases as the estimated noise level for a particular output value increases. Assuming that the noise levels for all N observations are equivalent then S_σ reduces to

$$S_\sigma = \frac{1}{\sigma_y^2}\sum_{i=1}^{N} \frac{1}{2}\left(f(\mathbf{x}_i|\mathbf{w},\mathcal{A}) - y_i\right)^2 \tag{A3}$$

where σ_y^2 is the variance for all observed output values. If we set β equal to the inverse of the variance $\sigma_y^2 = 1/\beta$ then S_σ further simplifies to $S_\sigma = \beta S$. Thus, β is interpreted as the inverse of the noise level present in the data. This interpretation can be further formalized by defining the likelihood of the parameter values

$$\Pr(D_N|\mathbf{w},\beta,\mathcal{A}) = \frac{1}{(\pi/\beta)^{N/2}}\exp(-\beta S). \tag{A4}$$

This is read as the likelihood of the network weights given a particular noise level (β), set of data (D_N), and network architecture ($\mathcal{A}$).[8] If a practitioner were certain of the noise level present in the data then the value of β may be set. If there is a large degree of uncertainty in β then Bayes theorem can be used to estimate it along with α.

Thus, far we have defined the prior and likelihood to the network weight parameters. The prior specified how smooth or complex we believe the network response should be. The likelihood defined how the network weights should be adjusted to estimate the observed data. Bayes Theorem states

$$posterior = \frac{likelihood \times prior}{marginal\,likelihood} \tag{A5}$$

thus, we can combine the prior and likelihood to determine the posterior distribution of the network weights. Assuming that we know what the values of α and β should be, the resulting form of Bayes Theorem is

$$\Pr(\mathbf{w}|D_N,\alpha,\beta,\mathcal{A}) = \frac{P(D_N|\mathbf{w},\beta,\mathcal{A})P(\mathbf{w}|\alpha,\mathcal{A})}{P(D_N|\alpha,\beta,\mathcal{A})} \tag{A6}$$

where $\Pr(\mathbf{W}|D_N, \alpha, \beta, \mathcal{A})$ is the posterior probability distribution of the network weights given a set of data (D_N), regularization hyperparameter values (α and β), and a network architecture ($\mathcal{A}$). The posterior is simply the product of the prior and likelihood divided by a normalizing constant ($\Pr(D_N|\alpha, \beta, \mathcal{A})$) that assures the posterior integrates to unity. This normalizing constant is referred to as the marginal likelihood of α and β, and is calculated by integrating the numer-

ator of the posterior with respect to all network weights

$$\Pr(D_N|\alpha,\beta,\mathcal{A}) = \int \Pr(D_N|\mathbf{w},\beta,\mathcal{A})\Pr(\mathbf{w}|\alpha,\mathcal{A})d^{N_W}\mathbf{w}. \quad \text{(A7)}$$

If optimal values for α and β are known then this normalizing constant can be dropped. The posterior for the NN weights then reduces to

$$\Pr(\mathbf{w}|D_N,\alpha,\beta,\mathcal{A}) \propto \exp(-(\alpha W + \beta S)). \quad \text{(A8)}$$

Thus, determining the maximum of the posterior is equivalent to finding the minimum of the regularization criterion of Eq. 2. However, optimal values for α and β have not yet been determined.

Appendix B: Optimization of α and β

Optimal values for α and β can be found by optimizing the Bayesian posterior for α and β. This can be formed from Bayes rule

$$\Pr(\alpha,\beta|D_N,\mathcal{A}) = \frac{\Pr(D_N|\alpha,\beta,\mathcal{A})\Pr(\alpha,\beta|\mathcal{A})}{\Pr(D_N|\mathcal{A})}. \quad \text{(B1)}$$

where the marginal likelihood in Eq. A7 is the likelihood here. The term $\Pr(D_N|\mathcal{A})$ is the marginal likelihood of the network architecture (another normalizing constant), and $\Pr(\alpha, \beta|\mathcal{A})$ is the prior joint probability distribution of α and β. MacKay[7] assumed that nothing was known about either of these parameters, and, thus, assigned a uniform prior to them $\Pr(\alpha, \beta|\mathcal{A}) \propto 1$. The posterior to the regularization parameters is then proportional to the normalizing constant in Eq. A6, which reduces to

$$\Pr(\alpha,\beta|D_N,\mathcal{A}) \propto \frac{1}{(\pi/\beta)^{N/2}(\pi/\alpha)^{N_W/2}} \int \exp(-M(\mathbf{w}|\alpha,\beta))d^{N_W}\mathbf{w}. \quad \text{(B2)}$$

Optimizing Eq. B2 with respect to α and β will result in optimal hyperparameter values. However, we must integrate $\exp(-M(\mathbf{w}|\alpha\ \beta))$ with respect to all of the network weights. This integral is not analytically tractable, however, it can be approximated via Markov Chain Monte Carlo methods,[29] or with a Gaussian approximation.[8] Markov Chain Monte Carlo methods will generally yield a more thorough solution, however, they are more computationally intensive and time-consuming. For this reason we will strictly use Gaussian approximations to calculate such integrals.

Once a set of weights is found that minimizes the regularization criterion M (for fixed values of the hyperparameters), a second-order Taylor series expansion can be made around the identified minima ($\boldsymbol{w}^{mp}$)

$$M(\mathbf{w}|\alpha,\beta) \approx M(\mathbf{w}^{mp}|\alpha,\beta) + \frac{1}{2}\left(\mathbf{w}-\mathbf{w}^{mp}\right)^T \mathbf{A}\left(\mathbf{w}-\mathbf{w}^{mp}\right) \quad \text{(B3)}$$

where $\mathbf{A}$ is the Hessian matrix with respect to the NN weights of the objective function ($\nabla^2 M$). This local approximation to the objective function can then be analytically integrated to approximate Eq. B2. The posterior for the regularization parameters is then approximated with

$$\Pr(\alpha,\beta|D_N,\mathcal{A}) \propto (\alpha)^{N_W/2}(\beta)^{N/2} \times \exp(-M(\mathbf{w}^{mp}|\alpha,\beta))(2\pi)^{N_W/2}|\mathbf{A}|^{-1/2}. \quad \text{(B4)}$$

By setting the derivative of this posterior equal to zero the optimal values for α and β can be determined, however, it is easier to take derivatives with respect to the log posterior

$$\log(\Pr(\alpha,\beta|D_N,\mathcal{A})) \approx \frac{N_W}{2}\log(\alpha) + \frac{N}{2}\log(\beta) - \alpha W(\mathbf{w}^{mp}) - \beta S(\mathbf{w}^{mp}) + \frac{N_W}{2}\log(2\pi) + \log\left(|\mathrm{A}|^{-1/2}\right) \quad \text{(B5)}$$

The Hessian can be further broken down in terms of α and β, ($\mathbf{A} = \alpha\mathbf{I} + \beta\nabla^2 S(\mathbf{w}^{mp} \mid \alpha\ \beta)$), where $\mathbf{I}$ is an identity matrix of size N_W. $\nabla^2 S(\mathbf{w}^{mp})$ is analytically tractable, however, here a Gauss-Newton approximation is used.[26] Setting the partial derivatives of Eq. B5 equal to zero, and solving for α or β yields new estimates for their most probable values. MacKay[7] showed that optimal regularization parameters could be found by using Eq. 3 and Eq. 4 to solve for α and β, respectively.

Appendix C: Optimizing β with an Informative Prior

When prior information is available for the noise levels of a system it can be included into the optimization of β. Here the prior formed in Eq. 7 is included into Bayes rule (Eq. B1) to yield a new posterior density for α and β

$$\log(\Pr(\alpha,\beta|D_N,\mathcal{A})) \approx \frac{N_W}{2}\log(\alpha) + \frac{N}{2}\log(\beta) - \alpha W(\mathbf{w}^{mp}) - \beta S(\mathbf{w}^{mp}) + \frac{N_W}{2}\log(2\pi) + \log\left(|\mathrm{A}|^{-1/2}\right) + \frac{N_R-1}{2}\log(\beta) - \frac{\beta\sum_{i=1}^{N_R}(y_i-\bar{y})^2}{2} \quad \text{(C1)}$$

Taking the derivative of Eq. C1 with respect to β yields

$$\frac{\partial\log(\Pr(\alpha,\beta|D_N,\mathcal{A}))}{\partial\beta} \approx \frac{N}{2\beta} - S(\mathbf{w}^{mp}) - \frac{1}{2\beta}(N_W + \alpha\mathrm{Trace}(\mathbf{A}^{-1})) + \frac{N_R-1}{2\beta} - \frac{\sum_{i=1}^{N_R}(y_i-\bar{y})^2}{2} \quad \text{(C2)}$$

Setting Eq. C2 equal to zero and solving for β yields the new optimal estimate for β shown in Eq. 8.

Manuscript received Dec. 13, 2006, and revision received Mar. 6, 2007.