Paper 1868-2018

Robust Tuning for Machine Learning

Alex Glushkovsky, BMO Financial Group

ABSTRACT

Tuning of models has become a very important topic. Machine learning models, especially neural nets, have many hyperparameters that may impact the outcome of the modeling process dramatically. Today, some machine learning products, such as SAS® Visual Data Mining and Machine Learning, include autotuning features. This paper discusses (1) different tuning criteria of the robust model, (2) possible approaches toward optimal tuning such as full factorial, screening, response surface methodology, Taguchi L-designs, and stochastic design of experiments, and (3) their practical implementation in SAS® Enterprise Miner™. Similar to the Nelder–Mead technique, optimal tuning may include iterative design of experiments extending the factor's levels that reach their limits at interim optimal points. Applying the Taguchi inner array for model hyperparameters and outer array for validation partition setups and randomizations creates the possibility to use dual response methodology to find a robust optimal solution acknowledging variance across outer arrays. Robust design of experiments allows for substantial subsampling of the training and testing datasets to accelerate the tuning process while controlling both signal and noise aspects. In addition to achieving model optimization, robust tuning provides insights on how model structure and hyperparameters influence the model performance. The latest allows learning by tuning.

INTRODUCTION

Modern trends in analytics have three distinct trajectories:

- Collection of huge information stored in databases. This trend is known as "Big Data"
- Powerful computational capabilities which includes parallel processing
- Broad libraries of descriptive and modeling algorithms

Thus, large datasets allows for fitting models with hundreds, if not thousands, of predictive variables and even incorporating their interactive and non-linear effects. Computational capabilities allow us not only to train very complicated models but to experiment with different model setups and get performance results for each setup within a reasonable timeframe. The latest trajectory allows for consideration of not just classical models such as regressions, but other approaches such as decision trees, random forest, gradient boosting, support vector machine, and neural nets.

These three trends create an opportunity to train better models. However, model development becomes a challenging process that not only addresses the best fitting of the model of a certain structure but tuning of the model structure and its hyperparameters as well. Machine learning models, especially neural nets, have many hyperparameters that may impact the outcome of the modeling process dramatically. Increased complexity of the learning process decreases the ability to guide this process by incorporating only expert knowledge. Thus, data scientists can assume an impact by changing some hyperparameters, but this assumption will probably only be directional and failing short to assume any interactive effects.

Imagine an ordinary case when the data preparation, which is usually the most puzzling part of machine learning, has been done. It means that the target and all potential inputs are defined and populated in the modeling dataset and now a data scientist is excited to start model training. Immediately, she or he will be facing some setup challenges without even starting modeling - at the data partitioning step. What split should be set between training and validation: 60/40 or 70/30? Should the random seed be set differently from the default value of 12345? Moving toward modelling, the number of questions concerning the setup of the modeling process and its hyperparameters rises dramatically.

Today, some machine learning products, such as SAS® Visual Data Mining and Machine Learning, include autotuning features. The "Automated Hyperparameter Tuning for Effective Machine Learning" has been discussed at SAS® Global Forum (Koch and *et al*, 2017). This excellent presentation introduces a great feature of the autotuning in SAS® Visual Data Mining and Machine Learning on SAS® Vija™ platform.

To succeed in the tuning process, three right "ingredients" should be defined:

1. The model fitting criterion that addresses the modeling objective
2. Design of Experiments (DOE) that chains reasonable combinations of values of the hyperparameters of interest
3. Optimization of hyperparameters within DOE ranges achieving the best result of the model fitting criterion

These three steps are not straightforward ones. Thus, it can be more than one modeling objective and even some conflicting ones. DOE needs to be balanced between the number of required trials and its resolution covering main, non-linear, and interactive effects. The optimization may not just require a simple selection of the best trial but it is a three-stage process of (1) fitting models based on DOE results, (2) simulation and constraint nonlinear programming, and (3) validation of the optimal solution. In addition, the definition of ranges of hyperparameters and number of levels in DOE is a very challenging step itself and may require iterative experiments similar to Nelder–Mead method (Nelder and Mead, 1965).

This paper addresses tuning that can be done using SAS® Enterprise Miner™.

Tuning actually covers two objectives: (1) to optimize hyperparameters toward the best model – this is a primary goal, and (2) to provide insights on how some elements of the model structure and hyperparameters influence model performance. The paper will focus on both serving aspects of tuning: Optimize and learn!

We are dealing with tuning in our everyday life (Figure 1). It is a mechanism that achieves the best fit. Concerning machine learning, tuning should be aligned with the model objectives. This seems as an evident requirement. However, in practice, this very logical point may be quite challenging and should be thoroughly defined.

Figure 1. Tuning Is Part of Our Everyday Activity

ROBUST MODEL CRITERIA

When training models, two types of questions arise:

- Is the model performance sound and does it cover its objectives?
- How stable is the model performance?

Answering these two types of questions supports optimization towards the best robust model.

The two above mentioned questions concerning model performance trigger subsequent questions:

- What type of model performance measurement should be selected?
- Is the model overfitted? How do you deal with the variance of the performance measure validating the model?

Therefore, two aspects of the modeling process should be addressed: (1) definition of the model performance criterion, and (2) incorporation of variance validating the model performance.

Examples of the former question can be illustrated for a problem having a binary target variable: “Should misclassification rate, Gini, K-S, or AIC criterion be used?”, or “What model is better, if the misclassification rate for one model is 5% and for another model it is 4%, while Gini is 83% and 80% correspondingly?”. Furthermore, since the misclassification rate combines both false positive and false negative cases, then: “Which one is more important?”.

Similar questions emerge when dealing with interval target variables: “Should the model performance be measured by R^2_{ADJ}, or by Mallows’ Cp, or simply by a value of the maximal residual?”.

It should be noted that SAS® Enterprise Miner™ has a “Model Comparison” node providing a wide range of model fit statistics grouped as: classification, data mining, and statistical measures.

The answer for “What appropriate performance measure should be selected?” depends on objectives of the model and its application. Considering predictive or discriminatory models, the immediate focus should be on the model’s primary goal and how the model will be used. Thus, for example, a model with a binary target variable may trigger the following question: “Is it for population classification into two groups?” or “Is it for discrimination of population elements along predictive scores?”. If the latter, then the subsequent question may arise: “Is it discrimination along the entire scoring population or most importantly of the first decile only?”. The challenge is that very often the difference between goals is not obvious.

The widely used Goal-Question-Metric paradigm (Basili and Rombach, 1988) or Improvement: Goal-Metric-Data approach based on Quality-Function-Deployment (QFD) matrixes (Glushkovsky, 2002) can be very helpful in guiding the selection of the model performance criteria. The latest is based on a matrix that has two dimensions that are related to the following questions: “What are the objectives?”, “How can they be measured?”, and the intensities of the relationships between these two parts. The development of such a matrix supports qualitative and quantitative analysis prioritizing and combining possible model performance criteria against the defined objectives.

Illustration of the QFD matrix is shown in Table 1 presenting model objectives versus measurement criteria. It should be noted that this relationship matrix is required to be tailored to a specific problem.

	How?		
What?	Criteria		
Model Objectives	Misclassification rate	K-S at first decile	Gini
Classification of objects	●	△	○
Discrimination (scoring) of population	○	△	●
Lift of the top decile	△	●	○

Relationships	
●	Strong
○	Medium
△	Weak

Table 1. Illustration of the QFD Relationship Matrix

The challenge of incorporating variance of measures validating the model arises when applying different validation partitions or k-fold cross-validations. The trivial and widely acknowledged approach is: “Use the average”. However, focusing not only on an average measure but on variances around the average allows for finding a more robust model. Here is a simple example: Let’s say one set of hyperparameters resulted in an average validation error of 8% and a standard deviation of 2% applying different data partitions. While another set of hyperparameters resulted in an average validation error of 9% and a

standard deviation of 0.5%. If we focus only on the average measure, then the first set has a better result. However, if considering variances, the second set provides a more significantly robust model with a slightly decreased average performance.

Consideration of the variance of the measure of interest and not just the average value is the key point of the dual response approach (Taguchi, 1986; Vining and Myers, 1990; Castillo and Montgomery, 1993).

Focusing on both the average and variance of the measure of interest and eventually applying dual response methodology can be helpful when dealing with a number of challenging issues of modeling, such as high volatility of small datasets, accelerated tuning by subsampling of large datasets, randomization issues of partitioning, cross validation (k-fold), and the existence of outliers.

This paper focuses on applying a dual response approach for tuning in machine learning.

DESIGN OF EXPERIMENTS

Finding the optimal hyperparameter setup is not a straightforward task considering the number of issues. Thus, there is *a priory* unknown dependency between the hyperparameter and the model performance criteria. For most hyperparameters, we do not know how sensitive that hyperparameter is, we do not know if there is a monotonic dependence or if there is a heavily non-linear impact that includes extreme point(s) within the factor's range, and we do not even know the direction of the impact in case of a monotonic impact. The factors can be of all types of variables: interval, ordered, nominal, or binary.

It should be noted that factors in the design of experiments are actually tuning hyperparameters of machine learning and in the paper, both terms will be used interchangeably.

In addition, we can barely assume the existence of interactive effects between parameters and predict their mutual effects. Taking into consideration all above mentioned issues, the practical way to find a solution is experimentation (Box *et al*, 1978; Montgomery, 2012). Basically, it can be one of the following approaches:

- Design of experiments within pre-defined ranges of factors
- Iterative design of experiments extending the factor's levels that reach their limits at interim optimal points
- Nelder–Mead technique which is a process of empirical optimization (Nelder and Mead, 1965)

The paper will focus on the former two approaches. The Nelder–Mead technique has a certain advantages that it does not require the planning of trials of design of experiments but still prerequisites the selection of hyperparamters, the size of the polytope, and definition of constraints. This technique may be very sensitive to the initial setup of hyperparameters values and the size of the polytope of $n + 1$ vertices having n factors. It is applicable to interval factors but is very challenging when dealing with categorical hyperparameters.

Finding the optimal setup of the model by tuning is a state-of-the-art process. For example, HP Regression model has 13 hyperparameters: Two-Factor Interactions, Polynomial Terms, Polynomial Degree, Supress Intercept, Use Missing as Level, Link Function for Logistic Regression, Optimization Options, Convergence Options, Selection Method, Selection Criterion, Stop Criterion, and Selection Options. In addition, dataset partitioning may include two factors: Partitioning Random Seed and Partitioning Data Set Allocations. Even by defining only two levels for each factor, the Full Factorial design will require 32,768 trials to tune a regression model!

Some interval hyperparameters have a very wide range of possible values. In some cases it is reasonable to apply unequal steps between levels. The latter may require their non-linear transformations during an analysis of experiments. For example, if we want to explore the factor levels 0.001, 0.01, and 0.1, it makes sense to apply LOG transformation in order to linearize factor levels.

Concerning DOE itself, it can be a wide spectrum of designs, including deterministic such as Full Factorial, Fractional Factorial, Screening, Response Surface Methodology (RSM), Taguchi L-designs, and stochastic ones.

FULL FACTORIAL

In the case when the number of hyperparameters is small (let say 2-5) and they are all interval variables, it is reasonable to perform a simple Full Factorial design 2^n, where each factor has two levels only. This design is an orthogonal and balanced one, and covers main linear and interactive effects. An advantage of the Full Factorial design is the easy programing by using Cartesian products between factors.

However, assuming a non-linear effect for an interval factor, the number of levels for that factor should be greater than or equal to three. It gives at least some information concerning the quadratic effects. For nominal factors, the number of levels in DOE is directed by the number of classes of interest of that factor.

SCREENING

Having a large number of hyperparameters, the first step may be the execution of the screening DOE (Plackett and Burman, 1946; Montgomery, 2012; http://www.jmp.com/support/help/Design_of_Experiments_Guide.shtml).

In these designs, only two levels of factors are applied. Screening designs, being very economical in terms of the required number of trials, are intended to find significant main effects only. Interactive effects cannot be estimated due to their confounding blend with the main effects.

Factors which survived after screening, may be included in the second stage DOE. The latest may be designed to estimate not only main linear effects but non-linear and interactive ones as well.

RESPONSE SURFACE METHODOLOGY

A more advanced approach of DOE is Response Surface Methodology (RSM). It allows for building regression models with interactive and non-linear (polynomial) effects (Box and Wilson, 1951). Practically however, the method can be applied to a limited number of factors considering 2^n+2n+1 number of required trials. Also, it is applicable to interval types of factors.

TAGUCHI L-DESIGN

Taguchi L-design (Ross, 1996) provides a wide range of DOE arrays for two or three levels from low resolution to high resolution. Low resolution designs only cover main effects that are confounded with possible interactions, while high resolution designs cover main, quadratic and pairwise effects. Also, it incorporates a framework distinguishing inner and outer arrays (see “TAGUCHI INNER AND OUTER DESIGN OF EXPERIMENTS” paragraph below).

STOCHASTIC DESIGN OF EXPERIMENTS

To reduce the number of DOE trials, Random Fractional Factorial DOE can be performed by applying a drop down rate for the inner trials while keeping a Full Factorial design for the outer factors, such as partition random seed and partition sets allocation. This approach is similar to “dropout” layers applied in some deep machine learning algorithms.

It can be easy implemented using the PROC SURVEYSELECT procedure specifying, for example, a dropout rate of 25% as RATE=%SYSEVALF(25/100). Therefore, it is not required to follow a predefined matrix but it cannot ensure orthogonal designs for inner arrays. The latest disadvantage is less critical when dealing with many factors and avoiding small dropout rates.

Random Full Factorial design can be a first stage followed by the fitting models and selecting important variables to be inputs for the second DOE stage. The second DOE stage usually has a higher resolution design allowing estimations of non-linear and interactive effects as well.

Alternatively, it is possible to perform random search or Latin Hypercube Sampling (LHS) (Koch and *et al*, 2017).

TAGUCHI INNER AND OUTER DESIGN OF EXPERIMENTS

Dr. Taguchi introduced a great idea of splitting the design of experiments into two arrays: inner and outer arrays consequently for two types of input variables, such as controllable and noise (Ross, 1996). The immediate candidates for the latest variables in machine learning are random seeds and proportions of the partitioning.

Moreover, applying the Taguchi inner array for model hyperparameters and outer array for validation partition setups creates a possibility to use dual response methodology to find a robust optimal solution acknowledging variance across outer arrays. In this case, the first response is the mean and the second response is the variance (standard deviation) aggregating measurements across the outer array (see Table 2).

DUAL RESPONSE METHODOLOGY

Implementing Taguchi design of experiments with inner and outer arrays allows for building of two models: (1) for the average and (2) for the variance (STD) of the measure of interest.

$$\hat{Y}_{MEAN} = f_m\left(\overrightarrow{H_I}\right) \quad (1)$$

$$\hat{Y}_{STD} = f_s\left(\overrightarrow{H_I}\right) \quad (2)$$

where $\overrightarrow{H_I}$ is a vector of inner hyperparameters

Combination of this approach with the Response Surface Methodology (RSM) and following non-linear optimization has been discussed in (Vining and Myers, 1990; Castillo and Montgomery, 1993; Lin and Tu, 1995).

Concerning tuning for machine learning, fitting two models (1) and (2) allows finding significant hyperparameters and their sensitivity. It supports learning about how model structure and hyperparameters influence the model performance. But most importantly, it can be used to find an optimal solution. For example, optimal tuning can be found by extensively simulating dual response results. For the latest case, it means interpolation and extrapolation within pre-defined hyperparameter ranges. Of course, the optimal solution that has been achieved by simulation requires post validation.

More specifically, simulation can be performed by applying uniform distributions within the ranges of hyperparameters that were defined in the DOE. Results of this simulation can be presented as a 2D scatterplot, where each point represents the average and the variance (STD) of the simulated trial.

Results of the dual response modeling should be validated facing two opposite but interdepending issues:

- Being robust, the models (1) and (2) may have large errors that can avert finding the optimal solution
- Being overfitted, the models (1) and (2) cannot provide accurate interpolations or extrapolations even within small margins deviated from the actual observations.

The shortfall of the dual response modeling approach is that the prediction error of the optimization criterion is a combination of errors of two models (1) and (2) and, therefore, the obtained result may be significantly off from the actual optimization point. Thus, by applying this approach, the post optimization validation is an absolutely necessary step to control this issue. To mitigate this issue, it is possible to fit a

direct model against the defined optimization criterion, such as Upper Confidence Limit (see equation 3 below). In this case, the estimation of the criterion may be more accurate but it is still a subject for further validation. Moreover, this possible improvement of accuracy grounds at the expense of losing individual visibility on underlying MEAN and STD factors.

Alternatively, the best tuning solution can be found within only performed trials of the DOE without simulation and without fitting models (1) and (2). In this case, the obtained results do not require further validation but may be suboptimal.

ILLUSTRATIVE EXAMPLE

The illustrative example concerning model tuning is based on SAS® Enterprise Miner™ High-Performance Data Mining (HPDM) nodes (Figure 2).

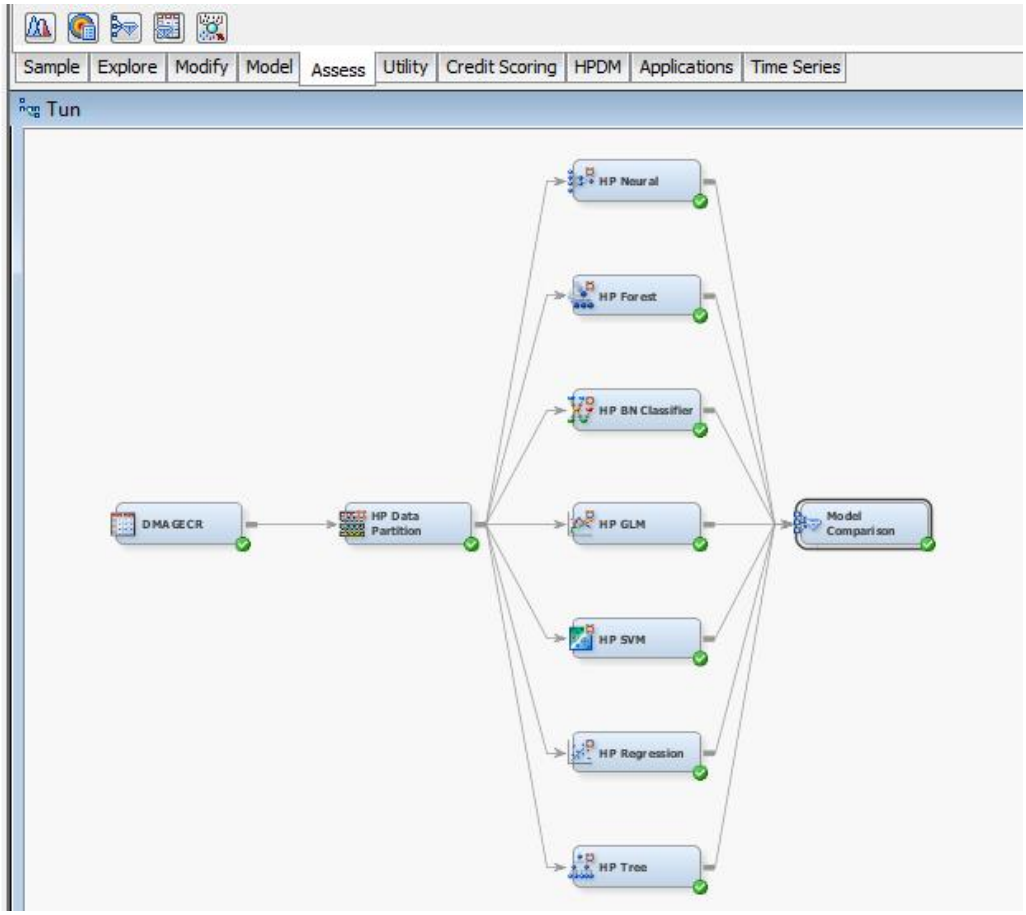


Figure 2. Diagram of the Illustrative Example in SAS® Enterprise Miner™

Dataset for modeling has been selected from the sample datasets for SAS® Enterprise Miner™ and is SAMPSIO.DMAGECR (German Credit Data Set). The dataset has 1,000 observations and 21 variables including a binary “good_bad” target. The description of the dataset can be found in (http://support.sas.com/kb/57/addl/fusion_57672_1_sampsio_data_sets.pdf).

All models have a default setup of hyperparameters, the partition random seed equals 12345, and allocations of 70% and 30% between training and validation datasets respectively.

ROC charts of the fitted models are shown in Figure 3.

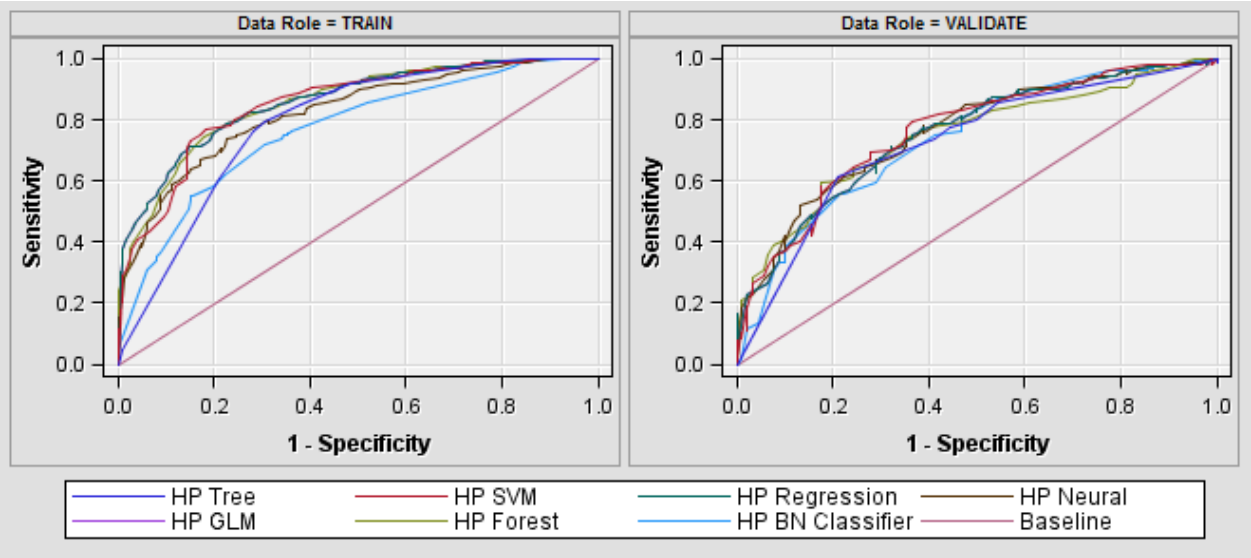


Figure 3. Comparison of ROC Charts of HP Models with Default Setups

It can be observed that most models are probably overfitted and/or the validation dataset is too small considering a significant decrease of the discriminatory power for the validation dataset.

In the illustrative example, validation misclassification rate has been selected as a model performance metric.

To consider variance of the models performance, three runs have been performed with different random seeds of the HP Data Partition node (11223, 12345, and 54321) having default setups of the modeling nodes. Fitting statistics for seven HP models are presented in Table 2 below.

	Validation: Misclassification Rate								
	Partition Random Seed			Dual Response		Optimization Criteria			
Model Description	11223	12345	54321	MEAN	STD	UCL	SN Ratio	Cpu	MSE
HP BN Classifier	0.257	0.233	0.247	0.246	0.012	0.261	12.190	0.127	0.0022
HP Forest	0.293	0.280	0.283	0.286	0.007	0.294	10.884	-1.708	0.0074
HP GLM	0.260	0.227	0.260	0.249	0.019	0.274	12.063	0.019	0.0028
HP Neural	0.270	0.233	0.240	0.248	0.020	0.273	12.101	0.038	0.0027
HP Regression	0.260	0.227	0.260	0.249	0.019	0.274	12.063	0.019	0.0028
HP SVM	0.260	0.240	0.247	0.249	0.010	0.262	12.075	0.036	0.0025
HP Tree	0.267	0.263	0.263	0.264	0.002	0.267	11.553	-2.502	0.0042

Table 2. Validation Misclassification Rates for Seven HP Models with Default Setups

Highlighted cells in Table 2 represent the best models, which are selected based on the minimum of the validation misclassification rate (VMISC) for each partition random seed. It can be concluded that the decision concerning the best model to be selected is very sensitive to the random seed value. Thus, for three different random seeds, four different models have been selected. In addition, volatilities of the misclassification rates across partition random seeds vary significantly ranging from 0.002 for HP Tree to 0.020 for HP Neural.

To visualize the above obtained results, Figure 4 represents validation misclassification rates for seven HP models across three random seeds of the data partition node.

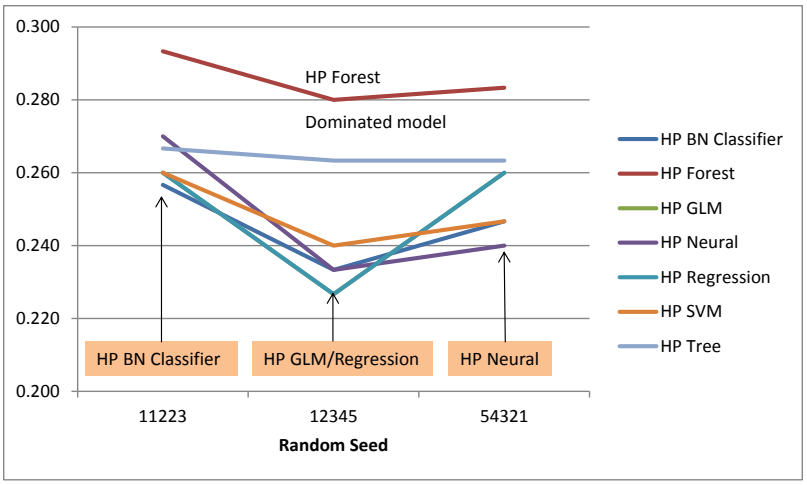


Figure 4. Validation Misclassification Rates of HP Models for Different Random Seeds of Data Partition

There are three dominated models: HP Forest is dominated by all other models, HP Tree is dominated by all but HP Neural and HP SVM is dominated by HP BN Classifier. If we limit our consideration to default settings only, then these dominated models should be excluded. However, there is no dominant model. It means that a consideration of dual response of mean and variance of validation misclassification rates across the outer array is required to select a best model.

DUAL RESPONSE OPTIMIZATION APPROACHES

Having dual responses leads to the following question: "How to find an optimal result?". It can be found based on unconstrained optimization combining both statistics to a single objective function, such as a confidence limit, or it can be done based on more complicated constrained optimization approaches.

Different optimization approaches of dual response problems have been discussed: Lagrangian constrained optimization (Vining and Myers, 1990), nonlinear programming solutions using generalized reduced gradient (GRG) (Castillo and Montgomery, 1993), and Mean Square Error (MSE) (Lin and Tu, 1995).

This paper will consider four different optimization criteria relating to mean and variance statistics:

- Upper Confidence Limit (UCL):

  $$UCL = MEAN + Z_{1-\alpha} * STD \quad (3)$$

  where the confidence level has been defined at 1-α=90%, $Z_{1-\alpha}$=1.28

- Taguchi Signal-to-Noise ratio (SN Ratio) for the smaller the better case (Ross, 1996):

  $$SN\ Ratio = -10 * LOG10(\frac{1}{n}\sum_{i=1}^{n} Y_i^2) = -10 * LOG10(MEAN^2 + STD^2) \quad (4)$$

- Capability Index concerning upper limit only (Montgomery, 2009), meaning the lower the misclassification rate the better:

  $$C_{pu} = \frac{USL - MEAN}{3 * STD} \quad (5)$$

  where the Upper Specification Limit (USL) in the illustrative example is set to 0.25

- Mean Squared Error (MSE) summarizing squared deviation of mean from target and variance (Lin and Tu, 1995):

$$MSE = (MEAN - Target)^2 + STD^2 \quad (6)$$

where Target value is set to 0.20

The goal of optimization for UCL and MSE is "the lower, the better" (i.e., minimum), while for SN Ratio and C_{pu} is "the higher, the better" (i.e., maximum).

Results of the above mentioned optimization criteria of the illustrative example are shown in Table 2.

The proportion of "bad" observations in the modeling dataset equals 0.30. The model which has very bad performance close to a "random" one will have a misclassification rate around that value for both the training and validation datasets and will almost be invariant to the random seed of the data partition. It means that variance across the outer array will be very small or zero. In this case, capability index C_{pu} gives a misleading signal to select a model that consistently reproduces the same results and does not stress the fact that these results may be bad. In this case, the optimization criterion prefers the robustness of the model over its underperformance. To mitigate this issue, a reasonable USL value should be selected. In the example, there are two very consistent but underperforming models: HP Tree and HP Forest. By setting USL equal to 0.25, which is below the population rate of 0.30 but practically achievable (other four models actually over performed that level), C_{pu} values for both models show negative values. It means that these two models are incapable of meeting a specified misclassification rate despite very high consistency.

As shown in Table 2, all four selected optimization criteria suggest that HP BN Classifier is the best choice for the SAMPSIO.DMAGECR dataset among other types of models with default setups. Of course, this consistency of results across different optimization criteria should not always be expected. In this paper, four criteria have been shown just for illustration. In real world cases, it is necessary to select an appropriate criterion according to the model objectives and to align it with the problem definition.

Applying dual response methodology, it allows for visualization of the obtained results on 2D scatterplot chart (Figure 5).

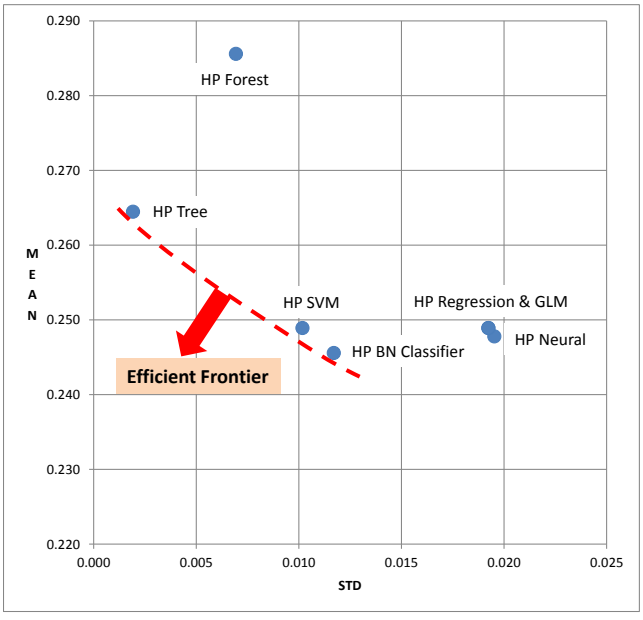


Figure 5. Dual Response Scatterplot (MEAN Vs STD) of Validation Misclassification Rate for Seven HP Models at Default Setups

The general optimization objective is a minimum of the validation misclassification rate. Observing variance of that measure by applying different random seeds of the data partition forms an efficient frontier on a dual response scatterplot. Obviously, the optimized location of the frontier is toward the bottom-left corner of the chart.

It can be observed that the efficient frontier in Figure 5 has a negative slope. It means that the lower the mean, the higher the variance of good candidates for an optimal solution. This is why the dual response optimization is important. For example, HP SVM and HP BN Classifier are both located on the efficient frontier. However, the HP BN Classifier has a better on average performance but a higher volatility compared to the HP SVM.

To make a decision, it is important to define an appropriate criterion aligned with the model objectives. In this paper, the UCL has been selected for illustrative purposes (see Equation 3 and Table 2).

IMPLEMENTATION OF ROBUST TUNING IN SAS® ENTERPRISE MINER™

Based on the UCL criterion presented in Table 2, the best model is the HP BN Classifier among six other HP models using default hyperparameters values. However, as an example, let us consider tuning of the HP Neural model. It should be noted that based on UCL criterion, this model is ranked only as number four out of seven trained models.

Practical implementation of tuning in SAS® Enterprise Miner™ can be done by executing the following basic steps:

1. Build a modeling diagram in SAS® Enterprise Miner™ where the default setting can be used
2. Export the batch processing code of the diagram by clicking on “Export Path as SAS Program” for the modeling node and then select “Run This Path” (Schubert, 2008)
3. Run the saved batch processing code. It can be done, for example, using SAS® Enterprise Guide®
   - Created output dataset NODEPROPS has information concerning properties of all nodes of the modeling diagram. It can be used when defining a DOE matrix
   - Created output dataset OUTFITDATA has information concerning model fit statistics. In the illustrative example, validation misclassification rate _VMISC_ has been selected as a model performance metric
4. Pack the batch processing code as a modular program
5. Change hyperparameters that have to be tuned to macro variables, such as

```
id= "HPPart"; property="RandomSeed"; value= &RandomSeed;
```

6. Assign values of the hyperparameters of each trial according to the defined DOE matrix
7. Run the modular program (4) and record results
8. Repeat steps 6-7 to complete DOE
9. Aggregate results across the outer array by calculating the average and the standard deviation statistics for each trial of the inner array
10. Find the optimal solution for inner hyperparameters of the dual response problem

Essentially, to perform tuning in SAS® Enterprise Miner™, two major elements are required: (1) creation of DOE matrix, and (2) running model fitting as a batch processing code where hyperparameter values are changing according to the defined DOE for each trial. Cycling through all DOE trials produces an output table that contains model fitting statistics. The produced output table can be used to find the optimal tuning solution.

RANDOM FRACTIONAL FACTORIAL DESIGN OF EXPERIMENT

The seven inner hyperparameters with mixed levels have been considered to tune the HP Neural model. The complete list of factors and their levels are presented in Table 3.

	Hyperparameters							
	Inner Factors							Outer Factor
	ARCHITECTURE	HIDDEN	InputStd	MAXLINKS	MISSASLVL	NUMTRIES	USEINVERSE	RandomSeed
Levels	5	3	3	2	2	2	2	3
Values	LAYER1	3	NONE	500	Y	2	Y	11223
	LAYER2	9	RANGE	1000	N	4	N	12345
	LOGISTIC	27	ZSCORE					54321
	LAYER1SKIP							
	LAYER2SKIP							

Table 3. Factors and Levels of the HP Neural Node Tuning DOE

The Random Fractional Factorial DOE process has the following steps:

(1) creation of the Full Factorial experiment across inner factors
(2) removal of trials that have invalid combinations of factors
(3) random sampling at a specified rate
(4) outer array definition.

In this example, the Full Factorial design for inner factors requires 5*3*3*2*2*2*2=720 trials.

The second step eliminates 96 trials by satisfying WHERE (ARCHITECTURE = '"LOGISTIC"') AND HIDDEN in ('"27"', '"9"') condition that has been specified to ensure compatibility of DOE trials with the valid combinations of hyperparameters (see "SAS® Enterprise Miner™ Reference Help" for more details).

Applying random subsampling at a rate of 25%, the number of trials has been reduced to 156.

The outer array is based on a single noise factor and three RandomSeed values of the HP Data Partition node. Therefore, the complete DOE requires 156*3=468 trials.

The fragment of the DOE including inner and outer arrays is presented in Table 4.

							Outer Array					
Inner Array							RandomSeed			Dual Response		Criterion
ARCHITECTURE	HIDDEN	InputStd	MAXLINKS	MISSASLVL	NUMTRIES	USEINVERSE	11223	12345	54321	MEAN	STD	UCL
LAYER2SKIP	27	NONE	1000	Y	2	Y	0.303	0.260	0.293	0.286	0.023	0.315
LAYER2	9	ZSCORE	500	Y	2	N	0.247	0.253	0.237	0.246	0.008	0.256
LAYER2	27	NONE	1000	Y	2	Y	0.700	[illegible]	0.297	0.452	0.217	0.730
LAYER2	9	NONE	500	N	2	Y	[illegible]	[illegible]	[illegible]	0.360	0.098	0.486
LAYER1	9	NONE	1000	Y	2	Y	0.[illegible]	[illegible]	0.[illegible]70	0.456	0.196	0.706
LAYER2SKIP	3	NONE	1000	N	2	Y	0.303	[illegible]	0.293	0.292	0.012	0.307
LAYER1SKIP	3	NONE	500	Y	4	N	0.237	0.247	0.227	0.237	0.010	0.249
LAYER1	27	RANGE	1000	Y	4	N	[illegible]	[illegible]	[illegible]	0.248	0.020	0.274
LAYER2	9	NONE	500	Y	2	N	[illegible]	[illegible]	[illegible]	0.300	0.000	0.300
LAYER2SKIP	3	ZSCORE	1000	Y	4	Y	0.303	0.277	0.320	0.300	0.022	0.328
LAYER1	3	NONE	500	Y	4	Y	0.303	0.433	0.390	[illegible]	[illegible]	0.460
LAYER1SKIP	9	ZSCORE	500	Y	2	Y	0.297	0.293	0.303	0.298	0.005	0.304
LAYER1	9	NONE	500	Y	2	N	0.300	0.300	0.300	0.300	0.000	0.300
LAYER1SKIP	3	RANGE	500	Y	4	Y	0.310	0.267	0.293	0.290	0.022	0.318
LAYER1	27	ZSCORE	1000	Y	4	N	0.230	0.233	0.230	0.231	0.002	0.234
LAYER2SKIP	9	RANGE	500	Y	4	N	0.253	0.237	0.237	0.242	0.010	0.255
LAYER2	27	RANGE	500	N	2	N	0.263	0.243	0.243	0.250	0.012	0.265
LAYER2	27	NONE	5[illegible]	[illegible]	[illegible]	[illegible]N	0.300	0.300	0.300	0.300	0.000	0.300
LOGISTIC	3	RANGE	10[illegible]	[illegible]	[illegible]	[illegible]N	0.257	0.247	0.237	0.247	0.010	0.259
LAYER1SKIP	9	RANGE	500	Y	2	N	0.250	0.233	0.243	0.242	0.008	0.253
LAYER2SKIP	27	RANGE	500	N	2	Y	0.290	0.277	0.293	0.287	0.009	0.298

Table 4. Fragment of DOE Matrix of the HP Neural Node Tuning

Performed Random Fractional Factorial DOE is the first stage in robust tuning. Of course, it is possible to select the best trial with respect to the defined optimization criterion UCL and end the tuning process. In our example, results of the best trial are shown in Table 4 as a highlighted row where UCL is minimal.

However, this result will most likely be suboptimal due to three reasons: (1) it is not a full but Random Fractional Factorial DOE, (2) the optimal point may be out of the specified factor's ranges, and (3) the optimal point may not be laying exactly on the levels of interval factors.

To improve the obtained results and to learn how DOE factors influence the optimization criterion, the next step may be dual response modeling.

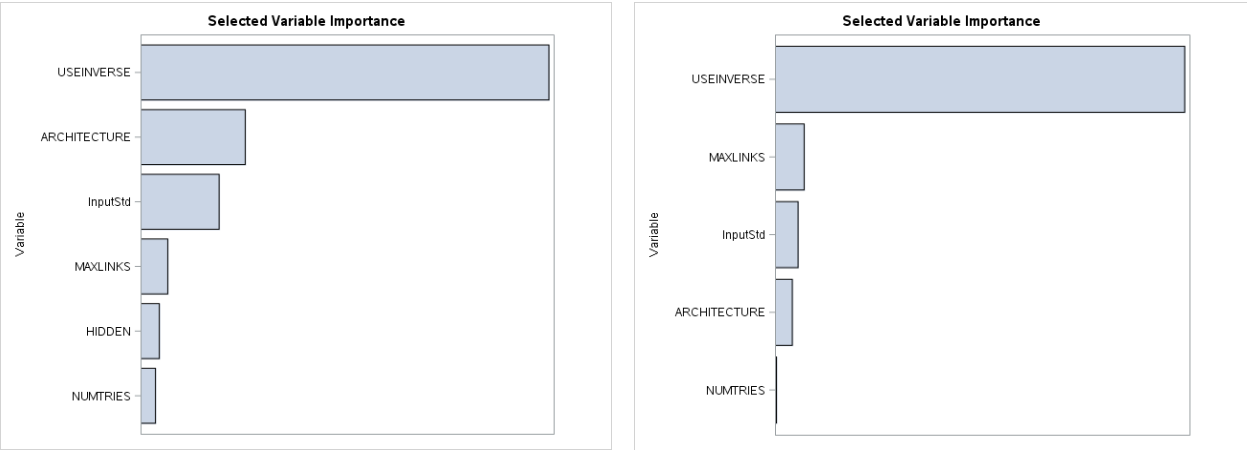
DUAL RESPONSE MODELING

The output Table 4 of the first stage can be used to train models (1) and (2) for MEAN and STD, respectively, or to fit a single model defining UCL as a target variable directly. Applying the latest case, however, will not lead to learning about the hyperparameters impact on MEAN and variance individually.

In this paper, two modeling approaches have been used: first, using SAS® Rapid Predictive Modeler (RPM) with the Intermediate modeling method and, second, using HP Neural model node.

The STD model has been trained on the transformed target variable using the PROBIT function with the following inverse by PROBNORM ensuring (0, 1) range of the predictive values.

Fitting RPM models (1) and (2) covers the "learn" aspect of tuning and provides insights on the importance of hyperparameters. It reveals that "USEINVERSE" is the most important factor for both models, while "MISSASLVL" has been rejected (Figure 6).



a) MEAN of VMISC Model (1) **b) PROBIT(STD) of VMISC Model (2)**

Figure 6. Importance of Hyperparameters of Fitted Models

The challenging issue of training dual response models, especially the STD one, is that there is usually no ability to perform partition into training and validation datasets due to already performed aggregation across the outer array. It means that overfitting may be a real problem that should be curbed by applying classical degrees of freedom paradigm. In addition, a mandatory validation of the predicted optimal result by applying fitted models (1) and (2) to an actual one is required.

Alternatively, HD Neural modeling with two hidden layers and TANH activation functions has been applied to train both (1) and (2) models (Figure 7).

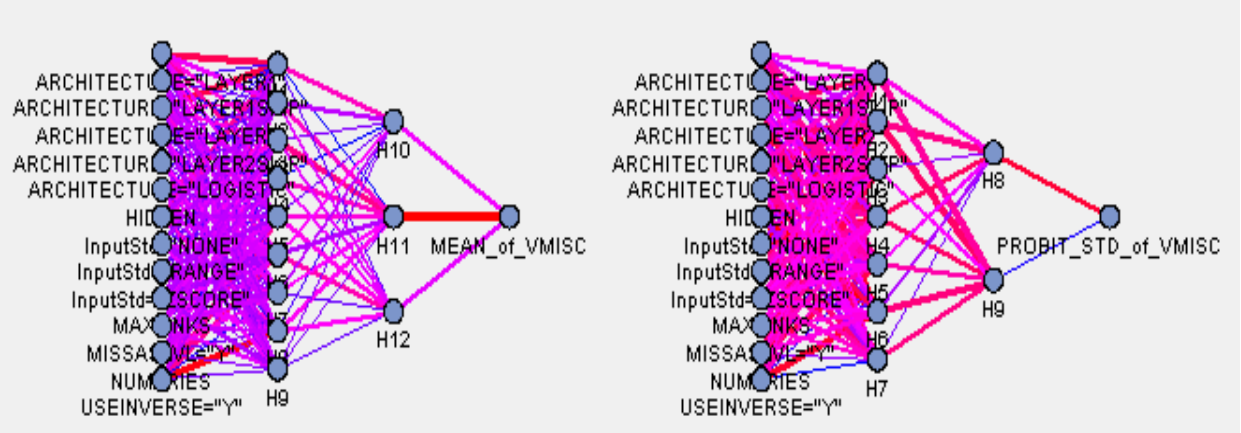


Figure 7. Link Graphs of the Fitted HP Neural (1) and (2) Models

It should be noted that by applying economical DOEs with insufficient coverage of the hyperparameter grid, the optimal tuning point in Table 4 will most likely be distanced from the best trial. Trying to mitigate this issue by fitting dual response models with subsequent optimization, the accuracy and stability of the trained models will be very questionable. It means that the selection of the DOE type is a crucial issue applying both tuning approaches.

DUAL RESPONSE SIMULATION

After fitting models (1) and (2), dual response simulation can be applied for survived hyperparameters where the number of levels of interval factors can be significantly increased. The motivation to apply simulation, based on the fitted models instead of the extended DOE of the first stage, is due to a significantly shorter scoring time that estimates MEAN and STD values instead of repeatable fitting of the original model.

To illustrate dual response simulation, a Full Factorial DOE with slightly extended hyperparameter ranges and significantly increased number of levels of interval factors has been applied (Table 5). Also hyperparameter “MISSASLVL” has been excluded as an unimportant one. The total number of the simulated trials is 64,360.

	Hyperparameters						
	Inner Factors						Outer Factor
	ARCHITECTURE	HIDDEN	InputStd	MAXLINKS	NUMTRIES	USEINVERSE	RandomSeed
Levels	5	29	3	15	5	2	3
Values	LAYER1	2-30	NONE	400-1100	2-6	Y	11223
	LAYER2	step = 1	RANGE	step = 50	step = 1	N	12345
	LOGISTIC		ZSCORE				54321
	LAYER1SKIP						
	LAYER2SKIP						

Table 5. Factors and Levels of the Simulated DOE

Results of dual response simulation applying fitted HD Neural (1) and (2) models to predict MEAN and STD, respectively, are shown in Figure 8.

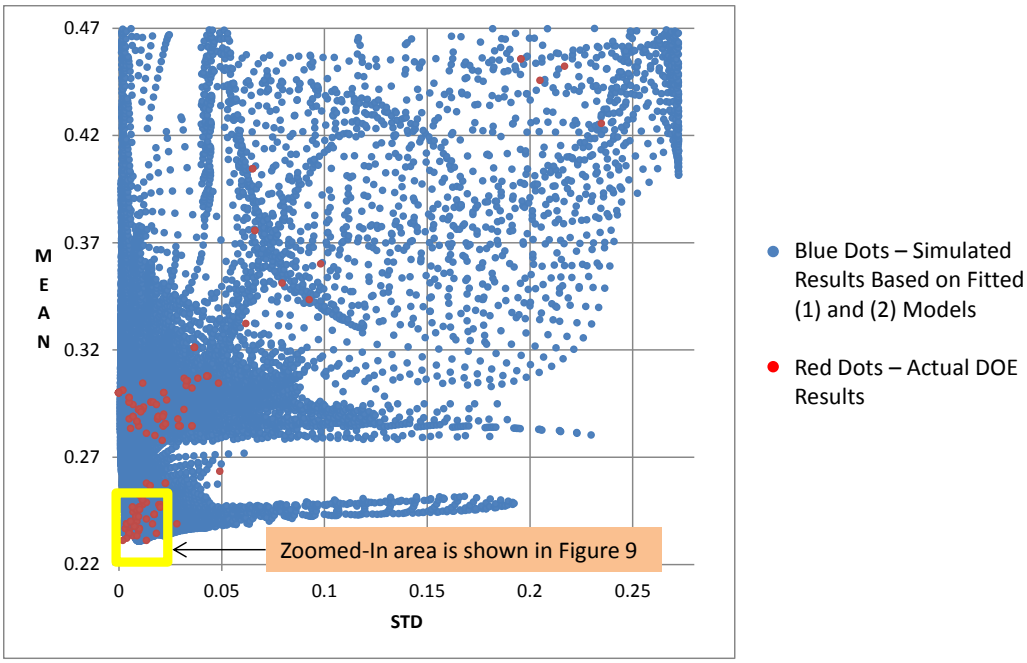


Figure 8. Dual Response MEAN versus STD Scatterplot

Observed patterns of blue dots in Figure 8 are due to the existence of input variables of the nominal types and discrete steps of the interval variables in the simulated DOE.

Small red points in Figure 8 are results of the Random Fractional Factorial experiment that has been used to fit models (1) and (2). Blue points represent simulated results based on the fitted models.

The most important bottom-left area of the chart has been zoomed in and shown in Figure 9.

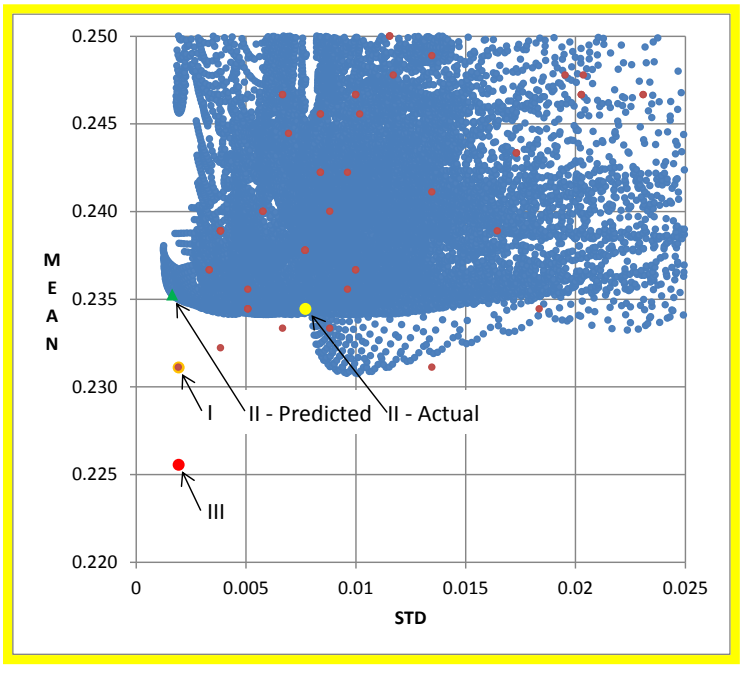


Figure 9. Zoomed In Bottom-Left Area of the Chart in Figure 8

The predicted optimal point “II” shown in Figure 9 as a green triangle is deviated from the actual result (yellow dot) mostly due to the STD prediction error. It is the expected result since modeling of the variance is a more difficult task than predicting the average. This observation may point out that defining only three levels of the outer partition random seed is not sufficient to confidently estimate the variance.

In Figures 8 and 9, some actual (small red) points are located outside of the predicted area (blue points). It is caused by the existence of model (1) and (2) fitting errors.

In the illustrative example, the simulation actually provides a slightly worst optimization result than one of the trials of the Random Fractional Factorial DOE (highlighted row in Table 4 and point “I” in Figure 9). It is due to the above mentioned deviations of the simulated values from the actual ones. That issue limits the application of the simulation approach. Nevertheless, dual response modeling with the following simulation may be a successful approach for some situations, especially when applying the Response Surface Methodology for interval factors (Vining and Myers, 1990; Castillo and Montgomery, 1993; Lin and Tu, 1995).

ITERATIVE DESIGN OF EXPERIMENTS

Similar to the Nelder–Mead technique (Nelder and Mead, 1965), optimal tuning may include an iterative design of experiments extending the interval factor’s levels that reach their limits at interim optimal points. It means extrapolation of DOE based on “educated” moves outside of the current borders of the experiment matrix.

Figure 10 illustrates arrays of two interval HIDDEN and NUMTRIES hyperparameters, where the blue points represent the initial DOE # 1 (Table 3), while the red points represent a series of iterative DOEs. The latest starts around the best “Tuning I” point of the initial DOE (highlighted trial in Table 4). It should be noted that “Tuning I” point is positioned on the upper-right corner of the DOE # 1 array. Iteratively extending DOE beyond the initial borders, a more preferable “Tuning III” point has been found. The new optimal “Tuning III” point is not located on a border of the DOE arrays indicating that at least a local optimum point has been pinpointed.

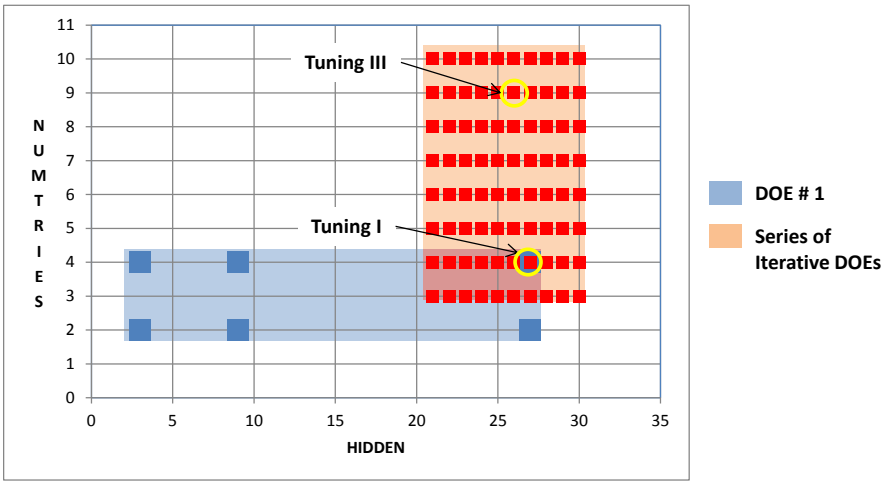


Figure 10. Mapping of Iterative DOE of Interval HIDDEN and NUMTRIES Hyperparameters

SUMMARIZED RESULTS OF HD NEURAL MODEL TUNNING

Results of the discussed three tuning approaches are presented in Table 6, where:

- Tuning I - performing Random Fractional Factorial DOE at a 25% reduction rate and selection of the best trial
- Tuning II - performing simulation based on fitted dual response models (1) and (2)

- Tuning III - performing iterative second stage high resolution DOEs around “Tuning I” optimal point

	Hyperparameters						Dual Response		Optimization Criteria			
Setup	ARCHITECTURE	HIDDEN	InputStd	MAXLINKS	NUMTRIES	USEINVERSE	MEAN	STD	UCL	SN	Cpu	MSE
Default	LAYER1	3	RANGE	1000	2	N	0.248	0.020	0.273	12.190	0.038	0.0027
Tuning I	LAYER1	27	ZSCORE	1000	4	N	0.231	0.002	0.234	15.611	3.272	0.0010
Tuning II - Predicted	LAYER1	23	ZSCORE	500	3	N	0.235	0.002	0.237	12.569	2.996	0.0012
Tuning II - Actual	LAYER1	23	ZSCORE	500	3	N	0.234	0.008	0.244	12.596	0.674	0.0012
Tuning III	LAYER1	26	ZSCORE	1000	9	N	0.226	0.002	0.228	12.935	4.234	0.0007

Table 6. Results of HP Neural Model Tuning

It can be observed that tuning significantly improves fitting of the HP Neural model compared to the default one. In the presented example, the UCL of the validation misclassification rate has been decreased from 0.273 to 0.228 (“Tuning III” point).

Tuned models demonstrate improvement of both average performance and robustness.

Dual response scatterplot visualizing the obtained results is shown in Figure 11. Clearly, tuned HD Neural models overperform all models with default setups.

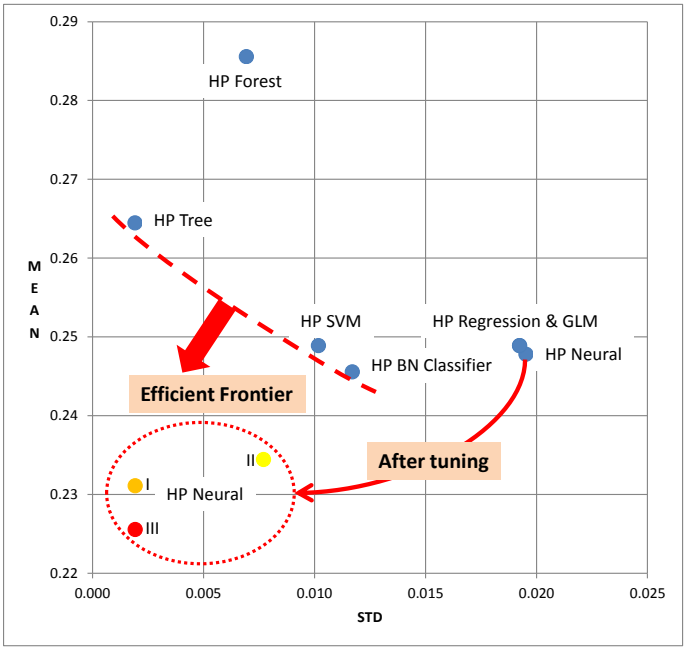


Figure 11. Dual Response Scatterplot (MEAN Vs STD) of Validation Misclassification Rate for Seven HP Models at Default Setup and Tuned HP Neural Models

Generally, the obtained optimization results should be tested by applying out-of-sample data if feasible, or by using different random seeds of the data partition. In case of significant deviation of the tested results from the expected ones, tuning using more inner and outer trials and increasing the sample sizes for modeling and validation should be considered. Of course, it will require more processing time.

The tuning process discussed in the paper is depicted in Figure 12.

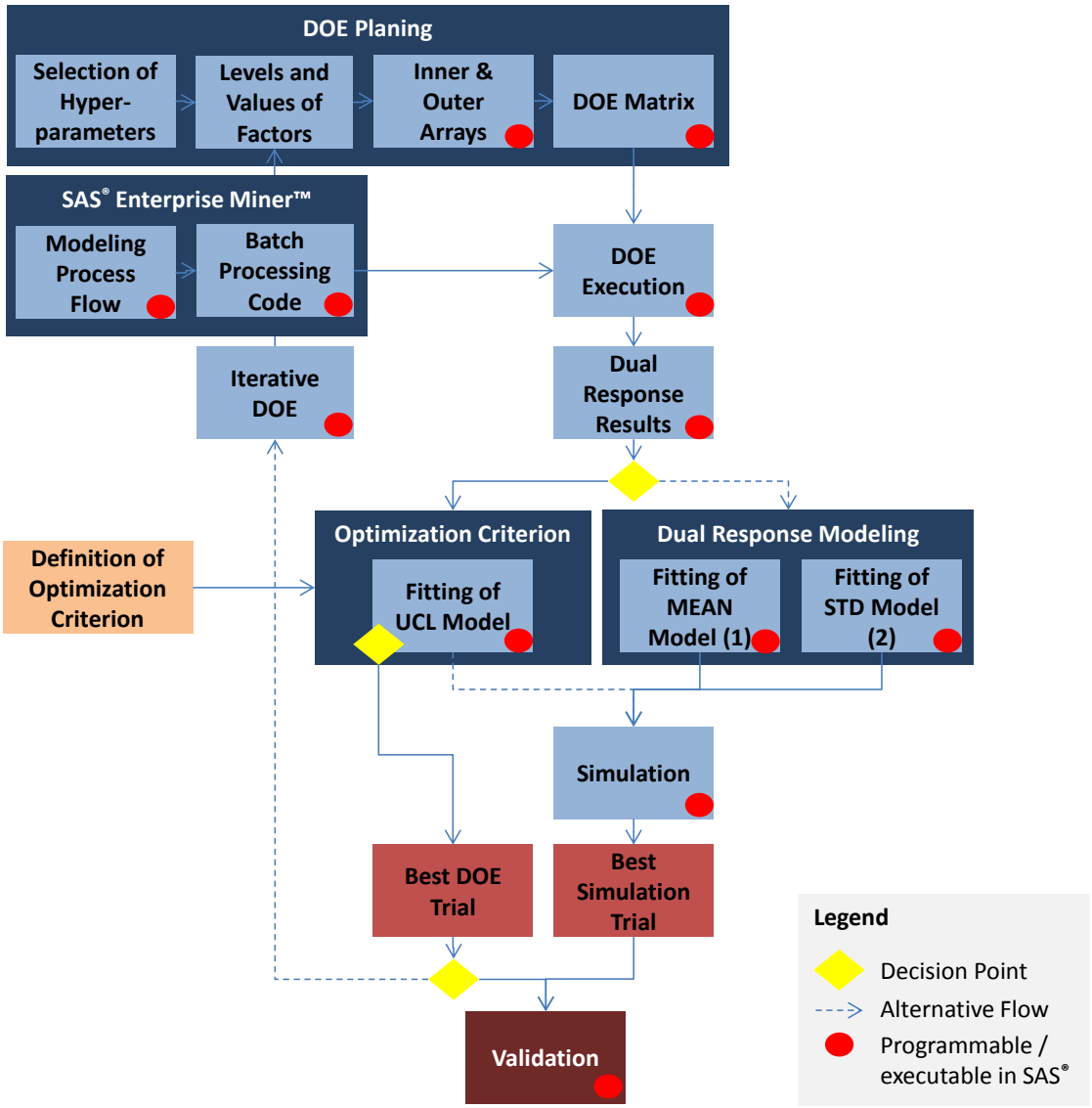


Figure 12. Diagram of Tuning Process

It should be noted that outer factors, such as a random seed of partition, most likely affect the parameters and even the structure of the model (for example, a decision tree). If so, then after tuning, it makes sense to train models for each trial of the outer array of the DOE while applying the optimized inner hyperparameters. The following ensemble of the trained models may be used to produce a final score (Maldonado and *et a*l, 2014).

ACCELERATED TUNING

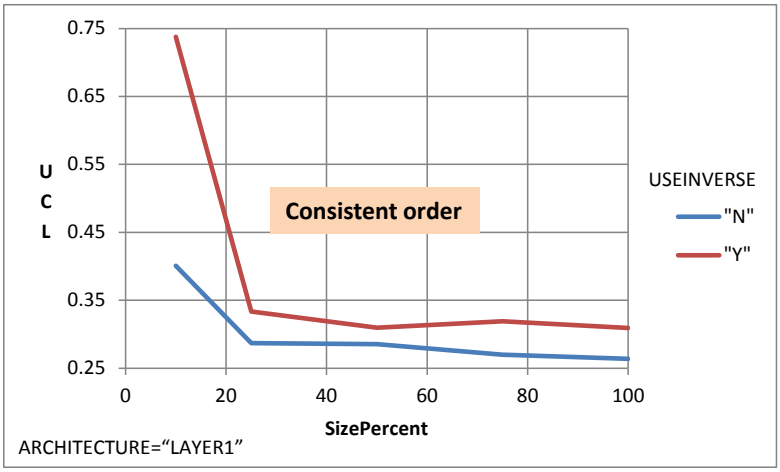
To accelerate tuning, three approaches and combinations of them can be explored:

- Implementation of economic design of experiments
- Dual response modeling with subsequent simulation
- Subsampling of the modeling dataset

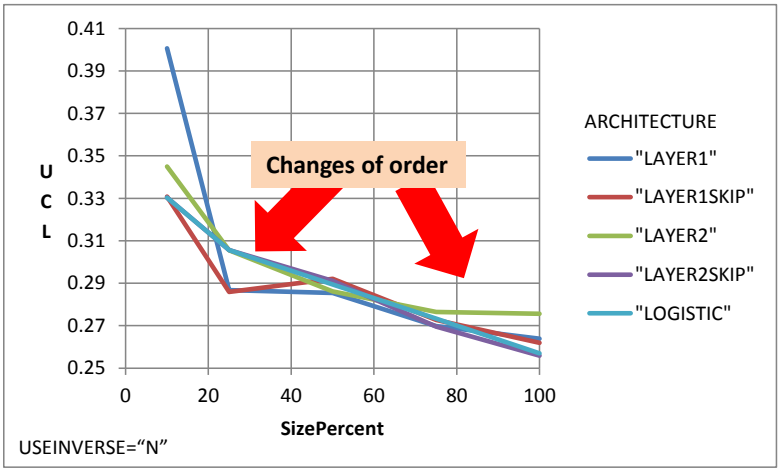
Two former approaches have been discussed earlier. Subsampling of the modeling dataset accelerates the tuning dramatically, but the question is “Would the optimized tuning solutions be the same by running it on different sizes of the modeling datasets?”.

To illustrate this concern, the following experiment has been performed. Random subsampling has been done using SAS® Enterprise Miner™ "Sample" node with the following setups of 10, 25, 50, 75, and 100% (complete dataset) of "Percentage" values. Then, HP Data Partition has been applied subsequently. The Full Factorial DOE has been applied running HP Neural model with changing values of the two top factors of the MEAN response model (Figure 6, a): "USEINVERSE" and "ARCHITECTURE". The experiment has been repeated three times with different values of the random seed (11223, 12345, and 54321).

Results of the experiment are shown in Figure 13.



a) For Different Values of Hyperparameter "USEINVERSE"



b) For Different Values of Hyperparameter "ARCHITECTURE"

Figure 13. Illustration of the Upper Confidence Limit of the Validation Misclassification Rate versus Subsampling Size of the Modeling Dataset

Considering the UCL as a measure of model performance, setting the "USEINVERSE" hyperparameter to "N" is the preferable setup regardless the subsampling sizes (Figure 13, a). Also, as expected, both curves of UCL have significantly elevated values for the smallest modeling dataset of 10% subsampling. It means that the modeling dataset of this size is too small to be a confidently used training model.

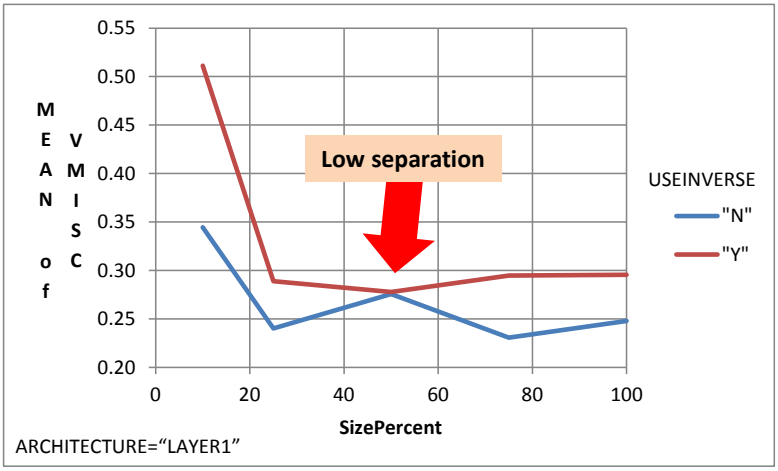
The optimal setup of the "ARCHITECTURE" hyperparameter is changing preferences along the subsampling sizes (Figure 13, b). It has a consistent order for sampling sizes between 50% and 100%,

while more aggressive subsampling produces a smaller modeling dataset and consequently unreliable results.

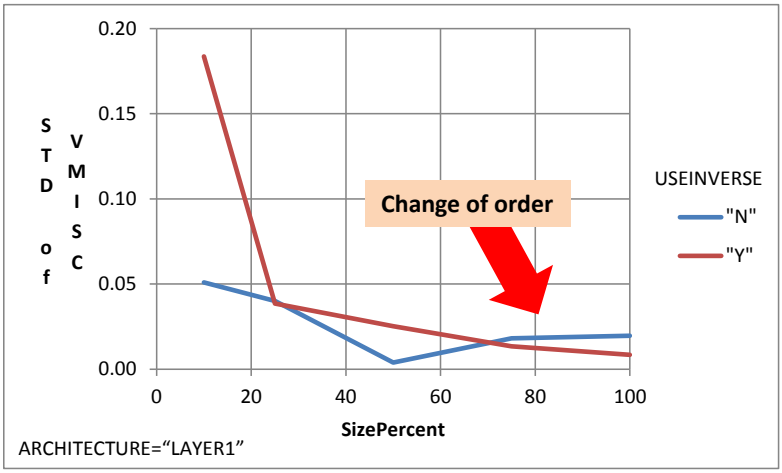
It should be noted that “USEINVERSE” is the most important factor for both fitted models (1) and (2), while “ARCHITECTURE” is the second important input of the MEAN model and only fourth important factor of the STD model (see Figure 6).

In this example, the higher the importance of the hyperparameter, the more consistent the results even when applying very aggressive subsampling. If the factor is not very important, then the optimization result may vary even for reasonable subsampling. Although the observed issue of a changing order affects less important hyperparameters, a suboptimal tuning setup may be an undesired result by applying subsampling.

To better understand the subsampling issue, let us consider the underlying statistics of the UCL criterion: the MEAN and STD (Figure 14).



a) MEAN of VMISC



b) STD of VMISC

Figure 14. Illustration of Validation Misclassification Rate Statistics versus Subsampling Size of the Modeling Dataset

It is interesting to observe that the optimization criterion UCL, incorporating both the average and the variance statistics of the model performance, is a more robust measure than just the average alone. Thus, the MEAN of VMISC has low separation at the 50% subsampling point, while the UCL criterion distinctively separates “USEINVERSE” levels along a very wide range of subsampling from 10% to 100%. In addition, it can be observed that the STD of VMISC has changing preferences of “USEINVERSE” levels along the subsampling sizes, where the most sensitive zone is for the smallest modeling dataset size (10%).

As it can be observed in the illustrative example above, focusing on both the average and variance allows for more robust results of the accelerated tuning process while decreasing training and validation datasets.

CONCLUSION

The starting point of tuning is defining the appropriate criterion of the model performance measure. Answering the following two questions relating to the QFD relationship matrix may be helpful in guiding this definition: “What are the objectives?” and “How can they be measured?”

Setting different random seeds of the data partition may significantly impact modeling results and the selection of the best model. Introduction of the Taguchi inner array for model hyperparameters and the outer array for validation partition and randomizations allows for robust tuning for machine learning focusing on both the average performance and its volatility.

Similar to the Nelder–Mead technique, optimal tuning may include an iterative design of experiments extending the levels of interval factors that reach their limits at interim optimal points.

Practical implementation of dual response tuning can be done using SAS® Enterprise Miner™ and the “Export Path as SAS Program” feature.

Accelerated tuning can be achieved using economic design of experiments, dual response modeling, and subsampling of modeling datasets. However, the obtained optimization results should be validated considering the stochastic nature of the accelerated tuning process.

In addition to achieving model optimization, dual response tuning provides insights on how model structure and hyperparameters influence the model performance and its robustness. It allows for learning by tuning.

REFERENCES

Basili, V. and Rombach, H. 1988. The TAME Project: Towards Improvement-Oriented Software Environment. *IEEE Transactions in Software Engineering*, Vol. 14, No. 6

Box, G. E. P., Hunter, W. G., and Hunter, S.J., 1978. *Statistics for experimenters: an introduction to design, data analysis, and model building*. Wiley

Box, G. E. P., and Wilson, K. B., 1951. “On the Experimental Attainment of Optimal Conditions”, *Journal of the Royal Statistical Society B*. Vol. 13, No. 1

Castello, E. and Montgomery, D. C., 1993. “A Nonlinear Programming Solution to the Dual Response Problem”, *Journal of Quality Technology*, Vo. 25, No. 3

Glushkovsky, A. 2002. Analytical Approach to Software Metrics Management, *Software Quality Professional*, ASQ, Vol. 4, No. 3

http://www.jmp.com/support/help/Design_of_Experiments_Guide.shtml

Koch, P., Wujek, B., Golovidov, O., and Gardner, S., 2017. “Automated Hyperparameter Tuning for Effective Machine Learning”, *Proceedings SAS Institute Inc*, SAS Paper SAS514-2017

Lin, D. K. J., and Tu., W., 1995. “Dual Response Surface Optimization”, *Journal of Quality Technology*, Vo. 27, No. 1

Maldonado, M., Dean, J., Czika, W., and Haller, S. 2014. “Leveraging Ensemble Models in SAS® Enterprise Miner™”, *Proceedings SAS Institute Inc*, SAS Paper SAS133-2014

Montgomery, D. C., 2009. Introduction to Statistical Quality Control. New York, Wiley

Montgomery, D. C., 2012. *Design and Analysis of Experiments*, 8th Edition, New York, Wiley

Nelder, J. and Mead, R. 1965. "A simplex method for function minimization". *Computer Journal*. Vol. 7, No. 4

Plackett, R. and Burman, J. 1946. "The Design of Optimum Multifactorial Experiments", *Biometrika* 33 (4)

Ross, P. J. 1996. *Taguchi Techniques for Quality Engineering*. McGraw Hill Professional

Schubert, S. 2008. Tailoring the Use of SAS® Enterprise Miner™, *Proceedings SAS Institute Inc*, SAS Paper 145-2008

http://support.sas.com/kb/57/addl/fusion_57672_1_sampsio_data_sets.pdf

Taguchi, G. 1986. *Introduction to Quality Engineering: Designing Quality into Products and Processes*. Kraus International Publications, White Plains, NY

Vining, G. G. and Myers, R. H. 1990. “Combining Taguchi and Response Surface Philosophies: A Dual Response Approach”, *Journal of Quality Technology*, Vo. 22, No. 1

DISCLAIMER

The paper represents the views of the author and do not necessarily reflect the views of the BMO Financial Group.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Alex Glushkovsky
BMO Financial Group
alex.glushkovsky@bmo.com