We consider a prediction model $g(x)$, where the input $x \in \mathbb{R}^d$ and the output $f \in \mathbb{R}$. Assume that the input follows a standard Gaussian distribution, i.e, $x \sim N(0, I_d)$, I_d is $d \times d$ identity matrix. We are interested in the probability $P(g(x) \geq \gamma)$, $\gamma \in \mathbb{R}$ is a threshold that triggers some certain rare events. We can run Monte Carlo to estimate $P(g(x) \geq \gamma)$, but when the probability is small, we might need a huge number of samples to obtain a reasonable estimation. Here, we study the use of importance sampling to largely reduce variance of the estimation.

A well-known IS scheme for Gaussian distribution is as following. We first find a dominating point a through optimization $a = \arg\max_x \{\phi(x) : g(x) \geq \gamma\}$. Then we use a Gaussian distribution with mean at a as the IS distribution. In the prediction model context, if we can formulate the optimization to be tractable, we have a IS scheme for the probability estimation problem.

1 Optimization over Neural Network

Here, we consider that the prediction model is a neural network. We suppose the neural network has L layers and for each layer, the number of neurons is $n_1, ..., n_L$. (As we consider that the output $g(x) \in \mathbb{R}$, we have $n_L = 1$.) We consider rectified linear unit (ReLU) for each neuron, i.e. the activation function for each neuron is $\max\{0, x\}$. The input of the jth neuron in layer i is weighted from the output of the previous layer by a vector $w_i^j \in \mathbb{R}^{n_{i-1}}$ and is added by a bias $b_i^j \mathbb{R}$. We use $s_i \in \mathbb{R}^{n_i}, i = 1, ..., L$ to represent the output of the ith layer. At the ith layer, given the output from the $(i-1)$th layer s_{i-1}, we have $s_i \in \mathbb{R}^{n_i}$, where the jth element of s_i is given by $s_i^j = \max\{{w_i^j}^T s_{i-1} + b_i^j, 0\}$.

The optimization problem for finding a dominating point is written as

$$\begin{aligned} \max_x \quad & \phi(x) \\ s.t. \quad & g(x) \geq \gamma \\ & g(x) \text{ is a neural network.} \end{aligned}$$

Now we plug the structure of the neural network into the constraints. We have

$$\begin{aligned} & s_L \geq \gamma \\ & s_i^j = \max\{{w_i^j}^T s_{i-1} + b_i^j, 0\},\ i = 1, ..., L,\ j = 1, ..., n_i \\ & s_0 = x. \end{aligned}$$

The constraints that contain max function require further handling. Given i, j, we decompose the equation as

$$\begin{aligned} s_i^j &\geq \max\{{w_i^j}^T s_{i-1} + b_i^j, 0\} \\ s_i^j &\leq \max\{{w_i^j}^T s_{i-1} + b_i^j, 0\}. \end{aligned} \tag{1}$$

The former inequality in (1) is further decomposed as

$$\begin{aligned} s_i^j &\geq {w_i^j}^T s_{i-1} + b_i^j \\ s_i^j &\geq 0. \end{aligned}$$

For the latter inequality in (1), since $\max(a, b) = \min(a, b) + |a - b|$, we rewrite as the following

$$s_i^j \leq \min\{{w_i^j}^T s_{i-1} + b_i^j, 0\} + |{w_i^j}^T s_{i-1} + b_i^j|.$$

We further have

$$\begin{aligned}
s_i^j &\leq {w_i^j}^T s_{i-1} + b_i^j + |{w_i^j}^T s_{i-1} + b_i^j| \\
s_i^j &\leq 0 + |{w_i^j}^T s_{i-1} + b_i^j|.
\end{aligned}$$

Now we introduce two dummy variables ${u_i^j}^+, {u_i^j}^- \in \mathbb{R}$, such that

$$\begin{aligned}
{u_i^j}^+ &\geq {w_i^j}^T s_{i-1} + b_i^j \\
{u_i^j}^- &\geq -{w_i^j}^T s_{i-1} - b_i^j \\
{u_i^j}^+, {u_i^j}^- &\geq 0.
\end{aligned}$$

Note that we always have ${u_i^j}^+ + {u_i^j}^- \geq |{w_i^j}^T s_{i-1} + b_i^j|$. Since ${u_i^j}^+, {u_i^j}^-$ does not appear in other constraints, we obtain a relaxation constraint by replacing $|{w_i^j}^T s_{i-1} + b_i^j|$ with ${u_i^j}^+ + {u_i^j}^-$. Here we rewrite the latter inequality in (1) for now, as following

$$\begin{aligned}
s_i^j &\leq {w_i^j}^T s_{i-1} + b_i^j + {u_i^j}^+ + {u_i^j}^- \\
s_i^j &\leq {u_i^j}^+ + {u_i^j}^- \\
{u_i^j}^+ &\geq {w_i^j}^T s_{i-1} + b_i^j \\
{u_i^j}^- &\geq -{w_i^j}^T s_{i-1} - b_i^j \\
{u_i^j}^+, {u_i^j}^- &\geq 0.
\end{aligned}$$

Note that this is yet a relaxation of the original problem. To make the reformulation equivalent to the original problem with regard to the optimal solution, we use a big-M approach. We choose a sufficiently large value for M. We add $M({u_i^j}^+ + {u_i^j}^-)$ into the objective function to force the value of ${u_i^j}^+ + {u_i^j}^-$ to be as small as possible, i.e., we obtain ${u_i^j}^+ + {u_i^j}^- = |{w_i^j}^T s_{i-1} + b_i^j|$. Therefore we reformulate the dominating point problem as

$$\begin{aligned}
\max \quad & \phi(x) + M\left(\sum_i \sum_j \left({u_i^j}^+ + {u_i^j}^-\right)\right) \\
s.t. \quad & s_L \geq \gamma \\
& s_i^j \geq {w_i^j}^T s_{i-1} + b_i^j && i = 1, ..., L,\ j = 1, ..., n_i \\
& s_i^j \geq 0 && i = 1, ..., L,\ j = 1, ..., n_i \\
& s_i^j \leq {w_i^j}^T s_{i-1} + b_i^j + {u_i^j}^+ + {u_i^j}^- && i = 1, ..., L,\ j = 1, ..., n_i \\
& s_i^j \leq {u_i^j}^+ + {u_i^j}^- && i = 1, ..., L,\ j = 1, ..., n_i \\
& {u_i^j}^+ \geq {w_i^j}^T s_{i-1} + b_i^j && i = 1, ..., L,\ j = 1, ..., n_i \\
& {u_i^j}^- \geq -{w_i^j}^T s_{i-1} - b_i^j && i = 1, ..., L,\ j = 1, ..., n_i \\
& {u_i^j}^+, {u_i^j}^- \geq 0 && i = 1, ..., L,\ j = 1, ..., n_i \\
& s_0 = x.
\end{aligned}$$

Now we consider to simplify the notations. We assume that $e_i = [1,...,1]^T \in \mathbb{R}^{n_i}$, $u_{sum} \in \mathbb{R}$,$W_i = [w_i^1,...,w_i^{n_i}] \in \mathbb{R}^{n_{i-1} \times n_i}$, $b_i = [b_i^1,...,b_i^{n_i}]^T \in \mathbb{R}^{n_i}$, $u_i{}^- = [u_i^{1-},...,u_i^{n_i-}]^T \in \mathbb{R}^{n_i}$, $u_i{}^+ = [u_i^{1+},...,u_i^{n_i+}]^T \in \mathbb{R}^{n_i}$. Then we simplify the problem as

$$\begin{aligned}
\max \quad & \phi(x) + Mu_{sum} \\
s.t. \quad & s_L \geq \gamma \\
& s_i \geq W_i{}^T s_{i-1} + b_i & i = 1,...,L \\
& s_i \geq 0 & i = 1,...,L \\
& s_i \leq W_i{}^T s_{i-1} + b_i + u_i{}^+ + u_i{}^- & i = 1,...,L \\
& s_i \leq u_i{}^+ + u_i{}^- & i = 1,...,L \\
& u_i{}^+ \geq W_i{}^T s_{i-1} + b_i & i = 1,...,L \\
& u_i{}^- \geq -W_i{}^T s_{i-1} - b_i & i = 1,...,L \\
& u_i{}^+, u_i{}^- \geq 0 & i = 1,...,L \\
& u_{sum} = \sum_{i=1}^{L} e_i{}^T (u_i{}^+ + u_i{}^-) \\
& s_0 = x.
\end{aligned}$$

The solution of the above optimization has the same optimal solution as the original problem. Note that the constraints are all linear, so as long as the objective is convex, this optimization is tractable. In our case, $\phi(x)$ is actually equivalent to $\|x\|^2$, which forms the problem as a quadratic optimization with linear constraints.

2 Importance Sampling with A Single Dominating Point

Here we consider the input $x \in [0,5]^2$. For the input $x = [x_1, x_2]$, we use the function

$$y(x) = (x_1 - 5)^3 + (x_2 - 4.5) + (x_1 - 1)^2 + 500 \tag{2}$$

to generate sample set $D = \{(X_n, Y_n)\}$. We use the sample set D to train the prediction models.

2.0.1 Neural Network Example

We generate 2,601 samples using mesh grid over the space $[0,5]^2$ for input values and using (3) to generate output values. We trained a neural network model with 3 layers. We use 100 neurons the 2 hidden layers and all neurons are ReLU. We refer to the output of the neural network at x as $g(x)$. We present the response surface of $g(x)$ in Figure 1. We use $\gamma = 500$ in this example and therefore the rare-event set in this case is $\{x : g(x) \geq \gamma\}$. The shape of the set is presented in Figure 2. We observe that the set should have a single dominating point.

3 Importance Sampling with Multiple Dominating Points

Again, we consider the input $x \in [0,5]^2$. For the input $x = [x_1, x_2]$, we use the function

$$y(x) = 10 \times e^{-\left(\frac{x_1-5}{3}\right)^2 - \left(\frac{x_2-5}{4}\right)^2} + 10 \times e^{-x_1{}^2 - (x_2-4.5)^2} \tag{3}$$

to generate sample set $D = \{(X_n, Y_n)\}$ for training the prediction models.

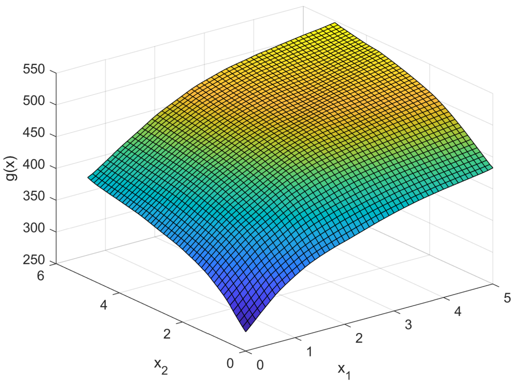

Figure 1: Response surface of the neural network model.

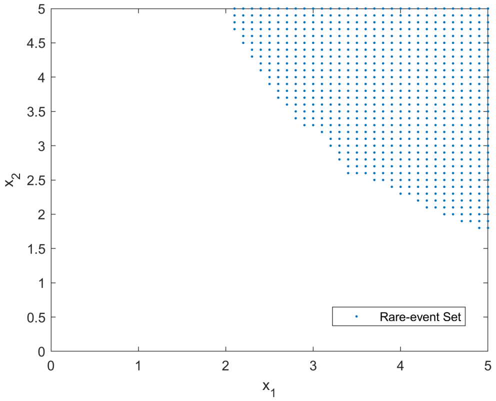

Figure 2: Rare-event set $\{x : g(x) \geq \gamma\}$ of the neural network model.