Suppose there are an arbitrary input $x \in \mathbb{R}^2$ and an arbitrary output $y \in \{0, 1\}$, and their probabilistic assumption is given by

$$p(\theta, x, y) = p(\theta)p(x)p(y|x;\theta), \tag{1}$$

where $q_\theta(x) := p(1|x;\theta)$ is modeled by the neural classifier with parameter θ. Also, assume a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ of N data points is given. Our goal is to make the Bayesian classifier for this dataset. Specifically, we want to learn

$$p(y|x) = \int p(y|x;\theta)p(\theta|\mathcal{D})d\theta \approx \frac{1}{N}\sum_{i=0}^{N-1} p(y|x;\theta_i), \tag{2}$$

where $\theta_0, ..., \theta_{N-1}$ are sampled from the posterior distribution $p(\theta|\mathcal{D})$ by Stein variational gradient descent (SVGD). From model assumption, we can say

$$\begin{aligned} p(\theta, \mathcal{D}) = p(\theta)p(\mathcal{D}|\theta) &= p(\theta)\prod_{i=0}^{N-1} p(x_i)p(y_i|x_i;\theta) && (3) \\ &= p(\theta)\prod_{i=0}^{N-1} p(x_i)q_\theta(x_i)^{y_i}(1 - q_\theta(x_i))^{1-y_i}, && (4) \end{aligned}$$

and the log posterior is

$$\log p(\theta|\mathcal{D}) = \log p(\theta) + \sum_{i=0}^{N-1} \{\log p(x_i) + y_i \log q_\theta(x_i) + (1 - y_i)\log(1 - q_\theta(x_i))\} - \log p(\mathcal{D}). \tag{5}$$

If we assume uniform prior $p(\theta)$ and ignore the terms that will not affect the gradient w.r.t. θ, the log posterior becomes

$$\log p(\theta|\mathcal{D}) = \sum_{i=0}^{N-1} \{y_i \log q_\theta(x_i) + (1 - y_i)\log(1 - q_\theta(x_i))\} + \mathit{constant} \tag{6}$$

Note that the term in summation is the same as the negative cross entropy which enables us to use cross entropy implementation in TensorFlow.