Minimsing the Loss Function for a Binary Classifier

Tom Hardy

July 8, 2025

1 Network Architecture

Consider a linear single-output layer neuron for a binary classifier network. The neuron computes some statistic $f(\mathbf{x}|\boldsymbol{w})$ given some training data $\mathbf{x}$ and an array of weights $\boldsymbol{w}$, which classifies some observation x_i into class A if $f(x_i|\boldsymbol{w}) > T$, else classifying to $\overline{A}$, where $\{A, \overline{A}\}$ represents a complete outcome of two mutually exclusive classes. Let there be N items of training data, of which N_A belong to A and $N_{\overline{A}}$ to $\overline{A}$, such that $N_A + N_{\overline{A}} = N$.

For the sake of Bayesian notation we define the likelihood ratio as

$$\texttt{lr}(\mathbf{x}) \equiv \frac{\mathbb{P}(\mathbf{x}|A)}{\mathbb{P}(\mathbf{x}|\overline{A})}, \tag{1}$$

the log-likelihood ratio as

$$\begin{aligned} \texttt{llr}(\mathbf{x}) &\equiv \log \texttt{lr}(\mathbf{x}) \quad (2)\\ &= \log \frac{\mathbb{P}(\mathbf{x}|A)}{\mathbb{P}(\mathbf{x}|\overline{A})}, \quad (3) \end{aligned}$$

and the Bayesian posterior, $\mathbb{P}(A|\mathbf{x})$, as

$$\begin{aligned} \mathbb{P}(A|\mathbf{x}) &\equiv \frac{e^{\texttt{llr}(\mathbf{x})+\rho}}{1 + e^{\texttt{llr}(\mathbf{x})+\rho}} \quad (4)\\ &= \sigma[\texttt{llr}(\mathbf{x}) + \rho] \quad (5) \end{aligned}$$

for a sigmoid σ and log prior odds ρ, where

$$\begin{aligned} \rho &\equiv \log \left[\frac{\mathbb{P}(A)}{\mathbb{P}(\overline{A})}\right] \quad (6)\\ &= \log \left[\frac{\mathbb{P}(A)}{1 - \mathbb{P}(A)}\right]. \quad (7) \end{aligned}$$

We can also define empirical priors for each dataset

$$p_A = \frac{N_A}{N},\ p_{\overline{A}} = \frac{N_{\overline{A}}}{N} \tag{8}$$

and we must use a network with some sigmoid activation function, which outputs some posterior

$$\mathbb{P}(A|\mathbf{x}, \boldsymbol{w}) = \sigma[f(\mathbf{x}, \boldsymbol{w}) + \hat{\rho}] \tag{9}$$

with some bias, $\hat{\rho}$, where

$$\hat{\rho} = \log \frac{p_A}{p_{\overline{A}}}. \tag{10}$$

It holds that we are free to pick **any** test statistic $f(\mathbf{x}|\boldsymbol{w})$, but for the most efficient network, we aim to minimise the loss $\mathscr{L}$: let us therefore find the form of $f(\mathbf{x}|\boldsymbol{w})$ which satisfies $\partial\mathscr{L}/\partial f = 0$.

2 Minimisation

For a binary classifier, we use the binary-cross-entropy loss function $\mathscr{L}_{BCE}$, which we define as

$$\mathscr{L}_{BCE} \equiv -\sum_{\mathbf{x}} \left[\frac{p_A}{N_A}\log\sigma + \frac{p_{\overline{A}}}{N_{\overline{A}}}\log(1-\sigma)\right], \tag{11}$$

for some sigmoid $\sigma \to \sigma[f(\mathbf{x}|\boldsymbol{w})]$ as convenient shorthand. Note that since some individual x_i element only satisfies one of each element, it becomes convenient to write the sum as

$$\mathscr{L}_{BCE} \equiv -\sum_{\mathbf{x}|A} \left[\frac{p_A}{N_A}\log\sigma\right] - \sum_{\mathbf{x}|\overline{A}} \left[\frac{p_{\overline{A}}}{N_{\overline{A}}}\log(1-\sigma)\right] \tag{12}$$

For large training data sets, these sums can be approximated as continuous integrals: with $\mathbb{E}_A[...]$ denoting the expected value across data $\mathbf{x} \in A$, we now approximate the loss as some continuous

$$\mathscr{L}_{BCE} \simeq -\Big[p_A\mathbb{E}_A\big[\log\sigma(f+\hat{\rho})\big] + p_{\overline{A}}\mathbb{E}_{\overline{A}}\big[\log[1-\sigma(f+\hat{\rho})]\big]\Big] \tag{13}$$

which holds under sufficiently large $\mathbf{x}$. In order to find the minimum, we use the method of calculus of variations; we assume there exists some optimum f^* at which $\partial\mathscr{L}/\partial f|_{f=f^*} = 0$ and explore around some pertubation $\gamma\varepsilon(\mathbf{x})$ for a small constant γ and arbitrary $\varepsilon(\mathbf{x})$, such that

$$\tilde{\mathscr{L}} \equiv \mathscr{L}[f^* + \gamma\varepsilon(\mathbf{x})]. \tag{14}$$

Then, by using the chain rule it follows that

$$\frac{\partial\tilde{\mathscr{L}}}{\partial\gamma} = -p_A\mathbb{E}_A\big[1-\sigma[f^*+\gamma\varepsilon+\hat{\rho}]\varepsilon\big] + p_{\overline{A}}\mathbb{E}_{\overline{A}}\big[\sigma[f^*+\gamma\varepsilon+\hat{\rho}]\varepsilon\big] \tag{15}$$

recalling that by construction $\partial\tilde{\mathscr{L}}/\partial\gamma|_{\gamma=0}$ (since f^* is the optimal solution), we then have

$$p_A\mathbb{E}_A\big[1-\sigma[f^*+\gamma\varepsilon+\hat{\rho}]\varepsilon\big] + p_{\overline{A}}\mathbb{E}_{\overline{A}}\big[\sigma[f^*+\gamma\varepsilon+\hat{\rho}]\varepsilon\big] = 0. \tag{16}$$

Explicitly writing out the expectation operator $\mathbb{E}_A[...]$ and factoring the pertubation $\varepsilon(\mathbf{x})$ then gives

$$\int_{\mathbf{x}} \left[-p_A \frac{1}{1+e^{f^*+\hat{\rho}}} \mathbb{P}(\mathbf{x}|A) + p_{\overline{A}} \frac{e^{f^*+\hat{\rho}}}{1+e^{f^*+\hat{\rho}}} \mathbb{P}(\mathbf{x}|\overline{A}) \right] d\mathbf{x} = 0. \tag{17}$$

These integrals are continuous, so it holds that

$$-p_A \frac{1}{1+e^{f^*+\hat{\rho}}} \mathbb{P}(\mathbf{x}|A) = p_{\overline{A}} \frac{e^{f^*+\hat{\rho}}}{1+e^{f^*+\hat{\rho}}} \mathbb{P}(\mathbf{x}|\overline{A}). \tag{18}$$

Simplifying further we have

$$e^{f^*+\hat{\rho}} = \frac{p_A}{p_{\overline{A}}} \frac{\mathbb{P}(\mathbf{x}|A)}{\mathbb{P}(\mathbf{x}|\overline{A})} \tag{19}$$

$$f^* + \hat{\rho} = \log \frac{p_A}{p_{\overline{A}}} + \log \frac{\mathbb{P}(\mathbf{x}|A)}{\mathbb{P}(\mathbf{x}|\overline{A})}. \tag{20}$$

Finally, recalling from the definition of the network's bias (equation (10)), we are left with

$$f^* = \log \frac{\mathbb{P}(\mathbf{x}|A)}{\mathbb{P}(\mathbf{x}|\overline{A})} \tag{21}$$

$$= \mathtt{llr}(\mathbf{x}) \ \blacksquare \tag{22}$$

To fully verify the Neyman-Pearson lemma, we must inspect the second derivative $\partial^2 \tilde{\mathscr{L}} / \partial \gamma^2$. We note that this can be expressed, in full

$$\partial^2 \tilde{\mathscr{L}} / \partial \gamma^2 = \int_{\mathbf{x}} \varepsilon^2(\mathbf{x}) \mathbb{p}(\mathbf{x}) \mu(\mathbf{x}, \gamma) d\mathbf{x} \tag{23}$$

where the probability density $\mathbb{p}(\mathbf{x})$ follows

$$\mathbb{p}(\mathbf{x}) = p_A \mathbb{P}(\mathbf{x}|A) + p_{\overline{A}} \mathbb{P}(\mathbf{x}|\overline{A}) \tag{24}$$

and we use $\mu(\mathbf{x})$ as shorthand for

$$\mu(\mathbf{x}, \gamma) = \left[\sigma(\mathtt{llr}(\mathbf{x}) + \gamma\varepsilon(\mathbf{x}) + \hat{\rho})\right] \left[1 - \sigma(\mathtt{llr}(\mathbf{x}) + \gamma\varepsilon(\mathbf{x}) + \hat{\rho})\right]. \tag{25}$$

We note that by definition ε^2 is positive; $\mathbb{p}(\mathbf{x})$ integrates to a positive scalar (by definition of a probability density); and by definition of the sigmoid, $\mu \in [0, 1]$: we conclude that $\partial^2 \tilde{\mathscr{L}} / \partial \gamma^2 > 0 : \forall \gamma \in \mathbb{R}$ verifying that **the log-likelihood is the optimal loss function for a suitably-sampled BCE neural network.**