On The Pairing Of The Softmax Activation And Cross–Entropy Penalty Functions And The Derivation Of The Softmax Activation Function

R. A. Dunne* & N. A. Campbell†

Abstract

It is suggested in the literature [2, 1] that there is a natural pairing between the softmax activation function and the cross–entropy penalty function. We clarify a reason for this pairing and give an improved derivation of the softmax activation function. In addition, we empirically compare some penalty/activation function pairs.

1 Introduction

The standard MLP model with two layers of adjustable parameters, p inputs, h hidden layer units and q output units, and no skip– or intra–layer connections, is described by

$$\mathrm{mlp}(x_i, \Upsilon, \Omega) = f_q(\Upsilon[1, \{f_h(\Omega x_i)\}^t]^t),$$

where Υ (of size $q \times h + 1$) and Ω (of size $h \times p + 1$) are the two matrices of adjustable parameters and $f_h : R^h \to (0,1)^h$ applies the same 1–variable "squashing" function to each of its coordinates. We take the "squashing" function to be the standard sigmoid $f_1(x) = 1/\{1 + \exp(-x)\}$.

We also have a set of training data $T = \{x_n, t_n\}_{n=1}^N$, where each x_n is a feature vector of length p, augmented by the addition of a 1 in the first coordinate position[1], and t_n is an encoding of the class label as a target vector of length q. t_{nk} is then the $(n,k)^{th}$ element of the target matrix.

For convenience we write $y = \Omega x$ and $y* = (1, \{f_h(y_1, \ldots, y_h)\}^t)^t$, the y vector being an argument to the function f_h, augmented by a 1 in the first coordinate position. This now forms the data input to the next, and in this case final, layer of the network. So we have $z = \Upsilon y*$ and

$$\mathrm{mlp}(x, \Upsilon, \Omega) = z* = f_q(z) = f_q(\Upsilon[1, \{f_h(\Omega x)\}^t]^t).$$

f_q will be either the logistic activation function, like f_h, or the softmax activation function,

$$f_q(k) = z_k^* = \frac{\exp(z_k)}{\sum_{k_1} \exp(z_{k_1})}.$$

We consider two penalty functions. One is the least squares penalty function

$$\rho_l = \sum_{n=1}^{N} \sum_{k=1}^{q} 1/2(t_{nk} - z_{nk}^*)^2,$$

and the other is the cross–entropy penalty function

$$\rho_c = \sum_{n=1}^{N} \sum_{k=1}^{q} t_{nk} \log(\frac{t_{nk}}{z_{nk}^*}).$$

Fitting the MLP model involves minimizing ρ, for which the derivatives with respect to the weights Υ and Ω are generally required. In the standard implementation these weights are initially assigned random values, chosen uniformly from a small interval, often $(-1, 1)$.

2 The "Natural" Pairing and Δ_k

We consider the case where z_k^* is given by the softmax activation function and ρ_c is the cross–entropy activation function. Then

$$\begin{aligned}
\frac{\partial \rho_c}{\partial z_k} &= \sum_{k_1=1}^{q} \frac{\partial \rho_c}{\partial z_{k_1}^*} \frac{\partial z_{k_1}^*}{\partial z_k} \\
&= \sum_{k_1=1}^{q} \frac{-t_{k_1}}{z_{k_1}^*} (z_{k_1}^* \delta_{k,k_1} - z_k^* z_{k_1}^*) \\
&= \sum_{k_1=1}^{q} -t_{k_1} (\delta_{k,k_1} - z_k^*) \\
&= \sum_{k_1=1}^{q} (t_{k_1} z_k^* - \delta_{k,k_1} t_{k_1}) \\
&= \left(\sum_{k_1=1}^{q} t_{k_1} \right) z_k^* - t_k \qquad (1) \\
&= \Delta_k \text{ say.}
\end{aligned}$$

*Rob Dunne is at the Victoria University of Technology, Victoria, Australia, email: dunne@matilda.vut.edu.au

†Norm Campbell is a Senior Principal Research Scientist at the CSIRO Mathematical and Information Sciences, Western Australia.

[1]The 1 supplies what is known as the "bias", which in some formulations of the MLP is supplied internally by the unit itself.

Hence, for $v_{jk} \in \Upsilon$ we can write

$$\frac{\partial \rho_c}{\partial v_{jk}} = \sum_{k_1=1}^{q} \frac{\partial \rho_c}{\partial z^*_{k_1}} \frac{\partial z^*_{k_1}}{\partial z_k} \frac{\partial z_k}{\partial v_{jk}}$$
$$= \Delta_k \frac{\partial z_k}{\partial v_{jk}}$$

and we can go on to calculate the derivatives of the other layers of the MLP.

As $\sum_{k_1=1}^{q} t_{k_1}$ will in general sum to one, Bishop gives (1) as $(z^*_k - t_k)$. He suggests that as:

- linear output units and a least squares penalty function;
- a two–class cross–entropy penalty function and a logistic activation function; and
- a multi–class cross–entropy penalty function and a softmax activation function

all give the same $\Delta_k = z^*_k - t_k$, there is a natural pairing of activation functions and penalty functions. When we use the natural pairing, we will always have $\Delta_k = z^*_k - t_k$.

However, it is easy to show that if we have a multi–class cross–entropy penalty function and a logistic activation function, we get the same Δ_k term. Taking z^*_k to be the output from a logistic activation function, we have

$$\frac{\partial \rho_c}{\partial z_k} = \sum_{k_1=1}^{q} \frac{\partial \rho_c}{\partial z^*_{k_1}} \frac{\partial z^*_{k_1}}{\partial z_k}$$
$$= \sum_{k_1=1}^{q} \frac{-t_{k_1}}{z^*_{k_1}} \frac{\partial z^*_{k_1}}{\partial z_k}$$

but as $\frac{\partial z^*_{k_1}}{\partial z_k} = 0$ if $k_1 \neq k$

$$\frac{\partial \rho_c}{\partial z_k} = \frac{-t_{k_1}}{z^*_{k_1}} \frac{\partial z^*_k}{\partial z_k}$$

and as $\frac{\partial z^*_k}{\partial z_k} = z^*_k - (z^*_k)^2$

$$\frac{\partial \rho_c}{\partial z_k} = z^*_k - t_k$$
$$= \Delta_k$$

and we have the same Δ_k term as before.

Unfortunately, the logistic activation function paired with the cross–entropy penalty term is not sensible. To see why, we plot the cross–entropy and least squares penalty functions (figures 1 and 2) for t and z^* in the region $[0, 1]^2$. For the least squares function, we can see that ρ has a minimum value of 0 whenever $z^* = t$. However, for the cross entropy error function, we have a minimum of $-e^{-1}$ at the point $(z^*, t) = (1, e^{-1})$ and, for a given non–zero target t_k, the function is minimized when $z^*_k = 1$. Hence the MLP will be returning 1 for all classes and thus only the additional constraint that $\sum_k z^*_k = 1$ makes the softmax activation function usable.

We can also consider pairing the least squares penalty function with the softmax activation function. Table 1 shows the Δ_k term for each of the possibilities. We note that the combination of least squares and softmax gives a more complex Δ_k.

Note that a combination of logistic outputs with a least–squares penalty function gives unbiased estimators of the posterior probability of class membership, $P(C|X)$, given that the MLP is of sufficient power to model the posterior probability to an arbitrary accuracy. Hence we have the asymptotic property that the outputs from a multi–class MLP with least squares and logistic functions will also sum to one. However, some simple experiments show that this convergence is too slow to be useful for reasonable sample sizes.

3 The Softmax Activation Function

The softmax activation function ensures that $\sum_k z^*_k = 1$, which is desirable in a *1 of q* classification, and allows us to use the cross–entropy error function. However, by modeling $P(x|C)$ we can give a better justification for the use of softmax activation function.

For a classification scheme using the sampling paradigm [7, 5], $P(C_i \mid x)$ is modeled as

$$P(C_i \mid x) = \frac{P(x \mid C_i) P(C_i)}{P(x)}$$

using Bayes' rule.

For a two–class problem, this becomes

$$P(C_1 \mid x) = \frac{P(x \mid C_1) P(C_1)}{P(x)}$$
$$= \frac{P(x \mid C_1) P(C_1)}{P(x|C_1) P(C_1) + P(x|C_2) P(C_2)}$$

which we can write as

$$= \frac{1}{1 + \exp\left\{ -\log\left[\frac{P(x|C_1)}{P(x|C_2)}\right] - \log\left[\frac{P(C_1)}{P(C_2)}\right] \right\}}. \quad (2)$$

Now if we are discriminating between two classes, with labels C_1 and C_2, and if we are making some distributional assumptions about $P(x \mid C_i)$, it is a standard procedure to base the test on the likelihood ratio,

$$\mathrm{LR} = \frac{P(C_1 \mid x)}{P(C_2 \mid x)}.$$

For computational reasons, we take minus the log of the likelihood and maximize this with respect to the parameters of the distribution $P(x \mid C_i)$.

$$\begin{aligned}\mathcal{L} &= -\log(\mathrm{LR}) \\ &= -\log\left[\frac{P(x \mid C_1)}{P(x \mid C_2}\right] - \log\left[\frac{P(C_1)}{P(C_2}\right].\end{aligned}$$

[5] comments that this mathematical step (2) will only be useful if the log likelihood has some convenient and tractable form.

However, if we start off with multiple classes and assume that $P(X|C_j)$ is a distribution from the exponential family of distributions[2], parameterized by (θ_j, ψ), we can derive the softmax activation function directly. Note that the distributions are assumed to have a common scale ψ.

$$\begin{aligned}P(C_k|X) =& \frac{P(X|C_k)P(C_k)}{\sum_{j=1}^{q} P(X|C_j)P(C_j)} \\ =& \frac{P(X|\theta_k,\psi)P(C_k)}{\sum_{j=1}^{q} P(X|\theta_j,\psi)P(C_j)} \\ =& \frac{\exp\left\{\frac{\theta_k^T X - b(\theta_k)}{a(\psi)} + c(X,\psi)\right\} P(C_k)}{\sum_{j=1}^{q} \exp\left\{\frac{\theta_j^T X - b(\theta_j)}{a(\psi)} + c(X,\psi)\right\} P(C_j)} \\ =& \frac{\exp\left\{\theta_k^T X - b(\theta_k) + \log[P(C_k)]\right\}}{\sum_{j=1}^{q} \exp\left\{\theta_j^T X - b(\theta_j) + \log[P(C_j)]\right\}} \qquad (3)\end{aligned}$$

Note that $\theta_k^T X - b(\theta_k) + \log[P(C_k)]$ is a linear combination of the variables with an offset or bias term and that (3) is the softmax activation function. This shows that modeling the posterior as a softmax function is invariant to a family of classification problems where the distributions are drawn from the same exponential family with equal scale parameters.

The logistic activation function is then recovered as a special case of softmax.

4 A Comparison of Least Squares and Cross–Entropy

This still leaves us with three possible estimators to consider:

- least–squares and the logistic activation function;
- least–squares and the softmax activation function;
- cross entropy and the softmax activation function.

The justifications for using these particular penalty functions are quite disparate. Both have been shown to result in z_k^* approximating $P(C_k|x)$ when used with the "one of q" target encoding. Least squares is justified on the basis of the Gauss–Markov theorem or intuitively, while cross–entropy is derived as a maximum likelihood (ML) estimator by modeling t_{nk} as a Bernoulli random variable. This gives us the result that, with some regularity conditions, if there exists an unbiased estimator which attains the Cramér–Rao minimum variance bound, then the ML estimator coincides with it. This does not seem very helpful with an MLP model, as the point estimate of the posterior probability $P(C|x)$ can only be shown to be unbiased when there are an arbitrary number of hidden–layer units [4]. The practice with very flexible models like the MLP is to introduce some bias in order to reduce the variance of the estimates [3]. In addition, the properties of Fisher efficiency that many ML estimators have has to be shown in each particular case [8].

Why then would we use one estimator rather than another? In particular, we would like to know how fast z_k^* converges to $P(C_k|x)$ (is it useful for reasonable–sized samples?) and how variable the two estimates are. There appears to be little guidance in the literature on these questions.

4.1 An Experiment

In the absence of any theoretical guidance, we consider a simple simulation. We take 100 observations from each of four Gaussian classes with means at the four symmetric points $\{\pm 1, \pm 1\}$ and unit variances and conduct 100 trials, generating independent samples for the training and testing cycles.

We consider two aspects here. One is the accuracy of the estimates of $P(C|x)$ and the other is the class–conditional error rates. The accuracy of $P(C|x)$ can be measured by the integrated difference between the estimated posterior probabilities and the exact probabilities

$$\sqrt{\sum_{i=1}^{4} \int (\hat{P}(C_i|x) - P(C_i|x))^2 dx}. \qquad (4)$$

We can calculate $P(C_i|x)$ as the distributions are known. We approximate (4) by

$$\sqrt{\sum_{i=1}^{4} \frac{1}{N} \sum_{n=1}^{N} \{\hat{P}(C_i|x_n) - P(C_i|x_n)\}^2};$$

[2] The exponential family of distributions includes the binomial, Poisson, negative binomial, gamma, Gaussian, uniform, geometric, exponential etc. See [8] [6], for a discussion of some of the properties.

this will only give a poor approximation but, as we are comparing the three estimators on the same data, this will allow us to rank them.

We can see from table 2 that the estimators are ranked, from worst to best, in the order:

1. logistic activation function and least squares;
2. softmax activation function and least squares;
3. softmax activation function and cross-entropy.

It would appear that the improvement is due to both the activation function and the penalty function. We next calculated the class–conditional error rates using the known distributions (table 3), and the empirical results (tables 4, 5 and 6). It can be clearly seen (by inspection or by a summary of the tables such as the trace) that the estimators are again ranked in the same order.

It would appear that even for a reasonably large sample (400 observations in 4 classes), the combination of the softmax activation function and the cross entropy penalty function gives the more accurate result.

5 Conclusion

We note that within the framework of function approximation [4], all that is required to show that the MLP is a universal approximator is that the activation functions be smooth, bounded, monotonic nonlinearities. However, it appears that modeling $P(x|C)$ as an exponential family distribution, and the target values as a Bernoulli random variable, thus recovering the softmax activation function and the cross–entropy penalty function, leads to more accurate results.

While the experiment described here is no substitute for a theoretical understanding of the properties of the three estimators, it does suggest that a probabilistic approach to the MLP model yields more accurate results. That this should be so in the case of the estimates of $P(C_i|x)$ is not surprising, as the logistic outputs fail to sum to one in some regions of the feature space. However, it is more surprising that the classification accuracy is also improved with the cross–entropy penalty function and the softmax function.

References

[1] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[2] J. S. Bridle. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems 2. Proceedings of the 1989 Conference*, pages 211–217, San Mateo, CA, 1990. Morgan Kaufmann.

[3] Stuart Geman, Elie Bienenstock, and René Doursar. Neural networks and the bias/variance dilemma. *Neural Computation*, 4(1):1–58, 1992.

[4] K. Hornik, M. Sinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–65, 1989.

[5] Michael I. Jordan. Why the logistic function? a tutorial on probabilities and neural networks. Computational Cognative Science Technical Report 9603, Massachusetts Institute of Technology, August 1995.

[6] C. R. Rao. *Linear Statistical Inference and Its Applications*. John Wiley & Sons, second edition, 1973.

[7] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[8] S. D. Silvey. *Stastical Inference*. Chapman and Hall, 1975.

	least–squares	cross-entropy
logistic	$(z_k^* - t_k)(z_k^* - (z_k^*)^2)$	$z_k^* - t_k$
softmax	$(z_k^*)^2 - t_k z_k^* - z_k^*(\sum_{k_1} t_{k_1} z_{k_1}^* - (z_{k_1}^*)^2)$	$z_k^* - t_k$

Table 1: *Δ_k for various pairings of penalty and activation functions.*

estimator	mean	variance
logistic activation function and least squares	4.621323	0.3146437
softmax activation function and least squares	4.003461	0.2315692
softmax activation function and cross-entropy	3.936804	0.1958714.

Table 2: *A comparison of the estimates of the quantity 4 for the three estimators.*

	1	2	3	4
1	0.7079	0.1335	0.1335	0.0252
2	0.1335	0.7079	0.0252	0.1335
3	0.1335	0.0252	0.7079	0.1335
4	0.0252	0.1335	0.1335	0.7079

Table 3: *The Bayesian class–conditional classification rates. The true class is shown down the side of the table and the ascribed class across the top of the table.*

	1	2	3	4
1	0.7001	0.1387	0.1336	0.0276
2	0.1450	0.6824	0.0243	0.1483
3	0.1350	0.0289	0.6999	0.1362
4	0.0344	0.1383	0.1398	0.6875

Table 4: *The class–conditional classification rates for the least squares error function and the logistic activation function. The true class is shown down the side of the table and the ascribed class across the top of the table.*

	1	2	3	4
1	0.7046	0.1376	0.1318	0.0260
2	0.1333	0.6981	0.0246	0.1440
3	0.1302	0.0265	0.7055	0.1378
4	0.0297	0.1343	0.1335	0.7025

Table 5: *The class–conditional classification rates for the cross–entropy penalty function and the softmax activation function. The true class is shown down the side of the table and the ascribed class across the top of the table.*

	1	2	3	4
1	0.7068	0.1397	0.1274	0.0261
2	0.1373	0.6958	0.0241	0.1428
3	0.1310	0.0261	0.7039	0.1390
4	0.0317	0.1349	0.1324	0.7010

Table 6: *The class–conditional classification rates for the least squares penalty function and the softmax activation function. The true class is shown down the side of the table and the ascribed class across the top of the table.*

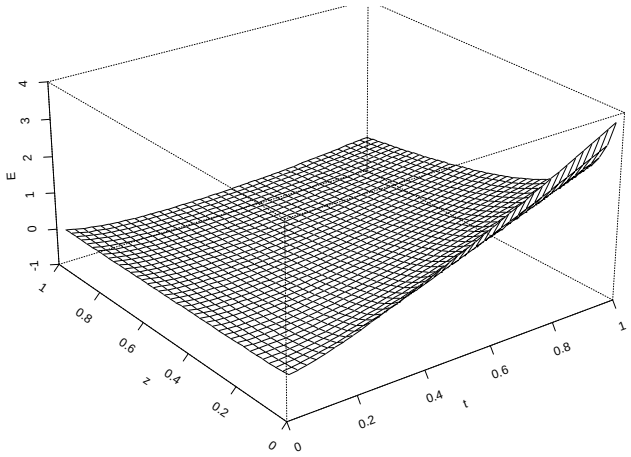

Figure 1: *The cross–entropy penalty function $\rho = t\log(t/z)$. The function has a minimum of $-e^{-1}$ at the point $(z^*, t) = (1, e^{-1})$ and, for a given non–zero target t_k, the function is minimized when $z_k^* = 1$.*

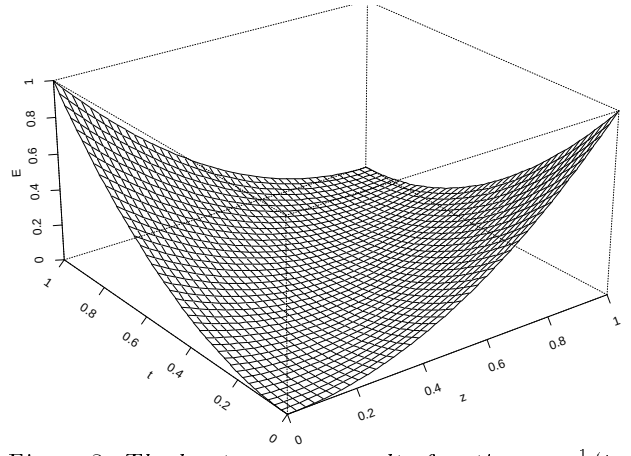

Figure 2: *The least squares penalty function $\rho = \frac{1}{2}(t - z)^2$ has a minimum value of 0 when $z = t$.*