10.7.2 Multiple Classes

Let us now generalize to $K > 2$ classes. We take one of the classes, for example, C_K, as the reference class and assume that

$$\log \frac{p(\boldsymbol{x}|C_i)}{p(\boldsymbol{x}|C_K)} = \boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}^o \tag{10.25}$$

Then we have

$$\frac{P(C_i|\boldsymbol{x})}{P(C_K|\boldsymbol{x})} = \exp[\boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}] \tag{10.26}$$

with $w_{i0} = w_{i0}^o + \log P(C_i)/P(C_K)$.

We see that

$$\begin{aligned} \sum_{i=1}^{K-1} \frac{P(C_i|\boldsymbol{x})}{P(C_K|\boldsymbol{x})} &= \frac{1 - P(C_K|\boldsymbol{x})}{P(C_K|\boldsymbol{x})} = \sum_{i=1}^{K-1} \exp[\boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}] \\ \Rightarrow \quad P(C_K|\boldsymbol{x}) &= \frac{1}{1 + \sum_{i=1}^{K-1} \exp[\boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}]} \end{aligned} \tag{10.27}$$

and also that

$$\begin{aligned} \frac{P(C_i|\boldsymbol{x})}{P(C_K|\boldsymbol{x})} &= \exp[\boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}] \\ \Rightarrow \quad P(C_i|\boldsymbol{x}) &= \frac{\exp[\boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}]}{1 + \sum_{j=1}^{K-1} \exp[\boldsymbol{w}_j^T \boldsymbol{x} + w_{j0}]}, \; i = 1, \ldots, K-1 \end{aligned} \tag{10.28}$$

To treat all classes uniformly, we can write

$$y_i = \hat{P}(C_i|\boldsymbol{x}) = \frac{\exp[\boldsymbol{w}_i^T \boldsymbol{x} + w_{i0}]}{\sum_{j=1}^{K} \exp[\boldsymbol{w}_j^T \boldsymbol{x} + w_{j0}]}, \; i = 1, \ldots, K \tag{10.29}$$

SOFTMAX

which is called the *softmax* function (Bridle 1990). If the weighted sum for one class is sufficiently larger than for the others, after it is boosted through exponentiation and normalization, its corresponding y_i will be close to 1 and the others will be close to 0. Thus it works like taking a maximum, except that it is differentiable; hence the name softmax. Softmax also guarantees that $\sum_i y_i = 1$.

Let us see how we can learn the parameters. In this case of $K > 2$ classes, each sample point is a multinomial trial with one draw; that is, $\boldsymbol{r}^t|\boldsymbol{x}^t \sim \text{Mult}_k(1, \boldsymbol{y}^t)$, where $y_i^t \equiv P(C_i|\boldsymbol{x}^t)$. The sample likelihood is

$$l(\{\boldsymbol{w}_i, w_{i0}\}_i|\mathcal{X}) = \prod_t \prod_i (y_i^t)^{r_i^t} \tag{10.30}$$

and the error function is again cross-entropy:

$$E(\{\mathbf{w}_i, w_{i0}\}_i|\mathcal{X}) = -\sum_t \sum_i r_i^t \log y_i^t \tag{10.31}$$

We again use gradient descent. If $y_i = \exp(a_i) / \sum_j \exp(a_j)$, we have

$$\frac{\partial y_i}{\partial a_j} = y_i(\delta_{ij} - y_j) \tag{10.32}$$

where δ_{ij} is the Kronecker delta, which is 1 if $i = j$ and 0 if $i \neq j$ (exercise 3). Given that $\sum_i r_i^t = 1$, we have the following update equations, for $j = 1, \ldots, K$

$$\begin{aligned}
\Delta \mathbf{w}_j &= \eta \sum_t \sum_i \frac{r_i^t}{y_i^t} y_i^t (\delta_{ij} - y_j^t) \mathbf{x}^t \\
&= \eta \sum_t \sum_i r_i^t (\delta_{ij} - y_j^t) \mathbf{x}^t \\
&= \eta \sum_t \left[\sum_i r_i^t \delta_{ij} - y_j^t \sum_i r_i^t \right] \mathbf{x}^t \\
&= \eta \sum_t (r_j^t - y_j^t) \mathbf{x}^t
\end{aligned}$$

$$\Delta w_{j0} = \eta \sum_t (r_j^t - y_j^t) \tag{10.33}$$

For the case of two classes we can write the likelihood of the data as

$$emp\ risk \ = \Pi_i p^{y_i} (1 - p)^{(1 - y_i)}$$

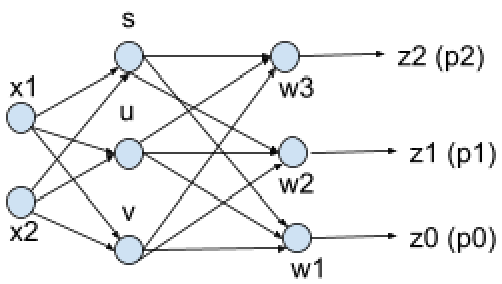
where p is the probability of class 1 and (1-p) is the probability of class 0 and i loops over my data points (xi,yi). Suppose p_0 is the probability of class 0 and p_1 is the probability of class 1. Then we can write the likelihood of the data

$$emp\ risk \ = \Pi_i p_1^{y_i} p_0^{(1 - y_i)}$$

Let c0 be the number of instances of xi with label 0 and c1 be the number of instances of xi with label 1. Then I can write the empirical risk as

$$likelihood \ = p_1^{c_1} p_0^{c_0}$$

Suppose we have a network with three nodes in the output layer (for three-way classification).



If we have three classes then the empirical risk becomes

$$likelihood = p_2^{c_2} p_1^{c_1} p_0^{c_0}$$

where p0 + p1 + p2 = 1 and c0 + c1 + c2 = n the total size of my training data.

We will convert the likelihood into the empirical risk by taking the negative log

$$emp\,risk = -\ log(p_2^{c_2} p_1^{c_1} p_0^{c_0}) = -\ c_2 log(p_2) - c_1 log(p_1) - c_0 log(p_0)$$

Each pj is the probability of the class j given the data and is given by the softmax function.

Suppose the outputs in the final layers are $z_0 = 1/(1 + e^{-w_1^T x})$, $z_1 = 1/(1 + e^{-w_2^T x})$, and $z_2 = 1/(1 + e^{-w_3^T x})$ which are also probabilities. Each zi is between 0 and 1.

This means I can write the empirical risk as

$$emp\,risk = f(w_1, w_2, w_3) = \ -\ c_2\, log(1/(1 + e^{-w_3^T x})) - c_1 log(1/(1 + e^{-w_2^T x})) - c_0 log(1/(1 + e^{-w_1^T x}))$$

To get the gradient I need the first derivatives with respect to each variable.

Let us keep the original form of the risk that loops over all datapoints.

$$emp\,risk\ = \Pi_j \Pi_i p_j^{y_i} p_0^{(1-y_i)}$$