function $H(\mathbf{p}, \mathbf{q})$ where

$$\begin{aligned} H(\mathbf{p}, \mathbf{q}) &= -(p_1 \ln(q_1) + \cdots + p_N \ln(q_N)) \\ &= -\sum_{k=1}^{N} p_k \ln(q_k). \end{aligned}$$

---

79 Theorem (Jacobian Matrix of Softmax)
The Jacobian matrix, $\frac{d\mathbf{p}}{d\mathbf{x}}$, of the softmax function $\mathbf{p} = f_{\mathsf{smax}}(\mathbf{x})$ is given by

$$\frac{d\mathbf{p}}{d\mathbf{x}} = \mathbf{D} - \mathbf{p}\mathbf{p}^\top$$

where $\mathbf{D}$ is a diagonal matrix with $\mathbf{p}$ along the main diagonal.

The softmax function $\mathbf{p} = f_{\mathsf{smax}}(\mathbf{x})$ is defined to be

$$f_{\mathsf{smax}}(\mathbf{x}) = \begin{pmatrix} \sigma_1(\mathbf{x}) & \cdots & \sigma_N(\mathbf{x}) \end{pmatrix}^\top$$

where

$$\sigma_k(\mathbf{x}) = \frac{e^{x_k}}{s} = p_k$$

where

$$s = e^{x_1} + e^{x_2} + \cdots + e^{x_N}.$$

Now for $i \neq k$,

$$\frac{\partial \sigma_k(\mathbf{x})}{\partial x_i} = \frac{\partial}{\partial x_i}\left(\frac{e^{x_k}}{s}\right) = -\frac{e^{x_i}\, e^{x_k}}{s^2} = -p_i p_k.$$

For $i = k$,

$$\frac{\partial \sigma_k(\mathbf{x})}{\partial x_i} = \frac{\partial}{\partial x_k}\left(\frac{e^{x_k}}{s}\right) = \frac{s e^{x_k} - e^{x_k} e^{x_k}}{s^2} = p_k - p_k p_k.$$

It thus follows that

$$\frac{d\mathbf{p}}{d\mathbf{x}} = \mathbf{D} - \mathbf{p}\mathbf{p}^\top$$

where $\mathbf{D}$ is a diagonal matrix with $\mathbf{p}$ along the main diagonal.

---

80 Definition (Cross-Entropy Loss Function)
Let $\mathbf{P}_{n\times N}$ be a one-hot-encoding of N classes. Then, the **cross-entropy loss function** $\ell(\hat{\mathbf{P}}, \mathbf{P})$ is defined to be

$$\ell(\hat{\mathbf{P}}, \mathbf{P}) = -\sum_{i=1}^{n}\sum_{k=1}^{N} p_{ik} \ln(\hat{p}_{ik}).$$

---

81 Example (Stochastic Gradient Descent)
The stochastic gradient descent algorithm has several advantages over the gradient descent algorithm algorithm. ,

- It is simplier.
- It is faster.
- It is less likely to over-fit.

---

82 Example (Stochastic Gradient Descent for Logistic Regression)
Recall that for logistic regression we have:

$$\mathbf{X}_{n\times(d+1)} \to \boxed{\mathbf{W}_{(d+1)\times N}} \to \hat{\mathbf{Y}}_{n\times N} \to \boxed{f_{\mathsf{smax}}} \to \hat{\mathbf{P}}_{n\times N}$$

$$\begin{aligned} \hat{\mathbf{Y}} &= \mathbf{X}\mathbf{W} \\ \hat{\mathbf{P}} &= f_{\mathsf{smax}}(\hat{\mathbf{Y}}) \end{aligned}$$

We choose the weight matrix $\mathbf{W}$ to minimize the cross-entropy loss function

$$\ell(\hat{\mathbf{P}}, \mathbf{P}) = \sum_{i=1}^{n} \ell_i(\hat{\mathbf{p}}_i, \mathbf{p}_i)$$

where

$$\ell_i(\hat{\mathbf{p}}_i, \mathbf{p}_i) = -\sum_{k=1}^{N} p_{ik} \ln(\hat{p}_{ik}).$$

Define the "gradient matrix"

$$\frac{d\ell}{d\mathbf{W}} = \begin{pmatrix} \frac{\partial \ell}{\partial W_{11}} & \cdots & \frac{\partial \ell}{\partial W_{1N}} \\ \vdots & \vdots & \vdots \\ \frac{\partial \ell}{\partial W_{(d+1)1}} & \cdots & \frac{\partial \ell}{\partial W_{(d+1)N}} \end{pmatrix}$$

Then we have the following types of iterations:

$$\mathbf{W} = \mathbf{W} - h\frac{d\ell}{d\mathbf{W}} \quad \text{gradient descent}$$

$$\mathbf{W} = \mathbf{W} - h\frac{d\ell_i}{d\mathbf{W}} \quad \text{stochastic gradient descent}$$

For large data sets, the gradient descent algorithm is slow because it takes a long time to compute a single gradient. It typically is much faster to compute an approximate gradient using only a small random sample of the data. This approach is called the **stochastic gradient descent** algorithm. The speed with which an approximate gradient can be computed enables many more gradient steps to be computed in the time it would take for a single full (exact) gradient vector to be computed. Typically, multiple approximate gradient vector steps reduce the loss function faster than a single accurate gradient vector step.

In the stochastic gradient descent algorithm, random samples (without replacement) are taken from the data set and used to compute the gradient vector of the loss function. The size of the random samples (called **minibatches**) is called the batch size. Once all the data in the data set has been used, we say one **epoch** has been completed. Typically, it takes multiple epochs for a stochastic gradient algorithm to converge. The optimal batch size is computing device dependent is typically determined using trial-and-error.

83 Example (Derivation of Stochastic Gradient for Logistic Regression) Show that

$$\frac{\partial \ell_i}{\partial w_{jk}} = -x_{ij}\sum_{r=1}^{N}\frac{p_{ir}}{\hat{p}_{ir}}\frac{\partial \hat{p}_{ir}}{\partial \hat{y}_{ik}}$$

where $\frac{d\hat{\mathbf{p}}_i}{d\hat{\mathbf{y}}_i} = \left(\frac{\partial \hat{p}_{ir}}{\partial \hat{y}_{ik}}\right)$ is the Jacobian matrix of the softmax function.

$$\begin{aligned} \ell_i &= -\sum_{r=1}^{N} p_{ir}\ln(\hat{p}_{ir}) \\ \frac{\partial \ell_i}{\partial w_{jk}} &= -\sum_{r=1}^{N}\frac{p_{ir}}{\hat{p}_{ir}}\frac{\partial \hat{p}_{ir}}{\partial w_{jk}} \end{aligned}$$

Now,

$$\frac{\partial \hat{p}_{ir}}{\partial w_{jk}} = \sum_{s=1}^{N}\frac{\partial \hat{p}_{ir}}{\partial \hat{y}_{is}}\frac{\partial \hat{y}_{is}}{\partial w_{jk}}.$$

Since

$$\frac{\partial \hat{y}_{is}}{\partial w_{jk}} = \begin{cases} 0 & s \neq k \\ x_{ij} & s = k \end{cases}$$

we have that

$$\begin{aligned} \frac{\partial \hat{p}_{ir}}{\partial w_{jk}} &= \frac{\partial \hat{p}_{ir}}{\partial \hat{y}_{ik}}\frac{\partial \hat{y}_{ik}}{\partial w_{jk}} \\ &= \frac{\partial \hat{p}_{ir}}{\partial \hat{y}_{ik}}x_{ij}. \end{aligned}$$

Therefore,

$$\begin{aligned} \frac{\partial \ell_i}{\partial w_{jk}} &= -\sum_{r=1}^{N}\frac{p_{ir}}{\hat{p}_{ir}}\frac{\partial \hat{p}_{ir}}{\partial \hat{y}_{jk}}x_{ij} \\ &= -x_{ij}\sum_{r=1}^{N}\frac{p_{ir}}{\hat{p}_{ir}}\frac{\partial \hat{p}_{ir}}{\partial \hat{y}_{jk}} \end{aligned}$$
