Exercise 1: SNE and Kullback-Leibler Divergence (50 P)

SNE is an embedding algorithm that operates by minimizing the Kullback-Leibler divergence between two discrete probability distributions p and q representing the input space and the embedding space respectively. In 'symmetric SNE', these discrete distributions assign to each pair of data points (i, j) in the dataset the probability scores p_{ij} and q_{ij} respectively, corresponding to how close the two data points are in the input and embedding spaces. Once the exact probability functions are defined, the embedding algorithm proceeds by optimizing the function:

$$\begin{aligned} C &= D_{\mathrm{KL}}(p \,\|\, q) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) \end{aligned}$$

where p and q are subject to the constraints $\sum_{i=1}^{N}\sum_{j=1}^{N} p_{ij} = 1$ and $\sum_{i=1}^{N}\sum_{j=1}^{N} q_{ij} = 1$. Specifically, the algorithm minimizes q which itself is a function of the coordinates in the embedded space. Optimization is typically performed using gradient descent.

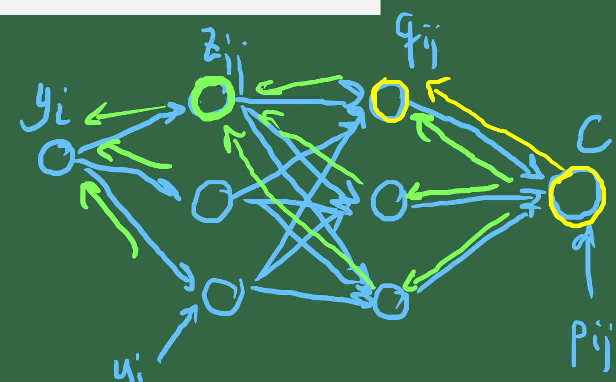
In this exercise, we derive the gradient of the Kullback-Leibler divergence, first with respect to the probability scores q_{ij}, and then with respect to the embedding coordinates of which q_{ij} is a function.

(a) *Show* that

$$\frac{\partial C}{\partial q_{ij}} = -\frac{p_{ij}}{q_{ij}}. \tag{1}$$

$$\frac{\partial C}{\partial q_{ij}} = \frac{\partial}{\partial q_{ij}}\left[\sum_m\sum_n p_{mn}\left[\log p_{mn} - \log q_{mn}\right]\right]$$

$$= \frac{\partial}{\partial q_{ij}} - p_{ij}\log q_{ij} = -\frac{p_{ij}}{q_{ij}}$$



(b) The probability matrix q is now reparameterized using a 'softargmax' function:

$$q_{ij} = \frac{\exp(z_{ij})}{\underbrace{\sum_{k=1}^{N}\sum_{l=1}^{N}\exp(z_{kl})}_{A}}$$

$$\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}$$

The new variables z_{ij} can be interpreted as unnormalized log-probabilities. *Show* that

$$\frac{\partial C}{\partial z_{ij}} = -p_{ij} + q_{ij}. \tag{2}$$

$$\frac{\partial C}{\partial z_{ij}} = \sum_m\sum_n \frac{\partial C}{\partial q_{mn}}\cdot\frac{\partial q_{mn}}{\partial z_{ij}} = \sum_{mn} -\frac{p_{mn}}{q_{mn}}\cdot\left[\frac{\delta(ij=mn)\cdot\exp(z_{mn})\cdot A - \exp(z_{mn})\exp(z_{ij})}{A^2}\right]$$

$$= -\frac{p_{ij}}{q_{ij}}\cdot\underbrace{\frac{\exp(z_{ij})}{A}}_{q_{ij}} + \underbrace{\sum_{mn}\frac{p_{mn}}{q_{mn}}q_{mn}}_{=1}\cdot q_{ij} = -p_{ij} + q_{ij}$$

(c) *Explain* which of the two gradients, (1) or (2), is the most appropriate for practical use in a gradient descent algorithm. Motivate your choice, first in terms of the stability or boundedness of the gradient, and second in terms of the ability to maintain a valid probability distribution during training.

stability / boundedness : Eq (2) is more stable + bounded b/c division by 0

maintain prob. dist : Eq (2) is better b/c softargmax always maintains a prob. dist.

(d) The scores z_{ij} are now reparameterized as

$$z_{ij} = -\|\mathbf{y}_i - \mathbf{y}_j\|^2$$

$$\frac{\partial\|x-y\|^2}{\partial x} = \frac{\partial\|y-x\|^2}{\partial x} = 2(x-y)$$

where the coordinates $\mathbf{y}_i, \mathbf{y}_j \in \mathbb{R}^h$ of data points in embedded space now appear explicitly. *Show* using the chain rule for derivatives that

$$\frac{\partial C}{\partial \mathbf{y}_i} = \sum_{j=1}^{N} 4\,(p_{ij} - q_{ij})\cdot(\mathbf{y}_i - \mathbf{y}_j).$$

$$\frac{\partial C}{\partial y_i} = \sum_j \frac{\partial C}{\partial z_{ij}}\cdot\frac{\partial z_{ij}}{\partial y_i} + \frac{\partial C}{\partial z_{ji}}\cdot\frac{\partial z_{ji}}{\partial y_i}$$

$$= \sum_j (-p_{ij}+q_{ij})\cdot(-2(y_i-y_j)) + (-p_{ji}+q_{ji})\cdot(-2(y_i-y_j))$$

$$= \sum_j 4\cdot(p_{ij}-q_{ij})\cdot(y_i-y_j)$$