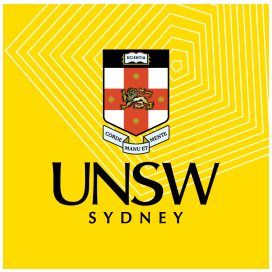
COMP9444
Neural Networks and Deep Learning
Term 2, 2024



Week 3 Tutorial: Probability, Generalisation and Overfitting (Sample Solution)

1. **Bayes' Rule**

   One bag contains 2 red balls and 3 white balls. Another bag contains 3 red balls and 2 green balls. One of these bags is chosen at random, and two balls are drawn randomly from that bag, without replacement. Both of the balls turn out to be red. What is the probability that the first bag is the one that was chosen?

Let B = first bag is chosen, R = both balls are red. Then

P (R | B) = (2/5)*(1/4) = 1/10

P (R | ¬ B) = (3/5)*(2/4) = 3/10

P (R) = (1/2)*(1/10) + (1/2)*(3/10) = 1/5

P (B | R) = P(R|B)*P(B) / P(R) = (1/10)*(1/2)/(1/5) = 1/4

2. **Entropy and KL-Divergence for Discrete Distributions**

   Consider these two probability distributions on the same space $\Omega = \{A, B, C, D\}$

$p = \langle \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8} \rangle$
$q = \langle \frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{1}{2} \rangle$

   (a) Construct a Huffman tree for each distribution p and q

---

The goal of Huffman coding is to assign a unique bit string to every possible event, such that more probable (i.e., more frequent) events have shorter code words (bit strings) while less probable events have longer code words. This is a desirable property because the average number of bits used to represent each event (and hence any message consisting of some sequence of these events) is reduced. (As an example, Morse code also has this property because the most common letters in English, "E" and "T," have the shortest possible codes: a single dot and a single dash.)

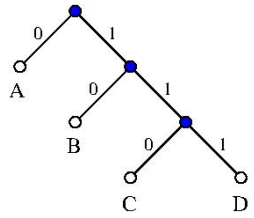


Figure 1: Huffman tree for distribution p

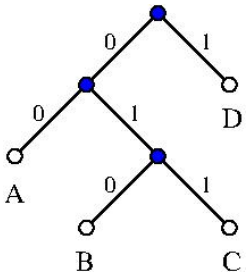


Figure 2: Huffman tree for distribution q

Huffman coding produces variable-length codes because different events are encoded using bit strings of different lengths.
Note: the answer is not unique; this is one possible tree in each case.

(b) Compute the entropy $H(p)$

H(p) = H(q) = $\frac{1}{2}(-log\frac{1}{2}) + \frac{1}{4}(-log\frac{1}{4}) + \frac{1}{8}(-log\frac{1}{8}) + \frac{1}{8}(-log\frac{1}{8})$

= $\frac{1}{2}(1) + \frac{1}{4}(2) + \frac{1}{8}(3) + \frac{1}{8}(3)$

= 1.75

(c) Compute the KL-Divergence in each direction $D_{KL}(q||p)$ and $D_{KL}(p||q)$. Which one is larger? Why?

$D_{KL}(p \parallel q) = \frac{1}{2}(2-1) + \frac{1}{4}(3-2) + \frac{1}{8}(3-3) + \frac{1}{8}(1-3) = 0.5$

$D_{KL}(q \parallel p) = \frac{1}{2}(1-2) + \frac{1}{4}(2-3) + \frac{1}{8}(3-3) + \frac{1}{8}(3-1) = 0.625$

$D_{KL}(q \parallel p)$ is larger, mainly because the frequency of D has increased from $\frac{1}{8}$ to $\frac{1}{2}$, so it incurs a cost of $3–1 = 2$ additional bits every time it occurs (which is often).

3. **Entropy, KL-Divergence and W_2 Distance for Bivariate Gaussians**

Consider two bivariate Gaussian distributions p and q (see figure).

q has mean $\mu_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and variance $\sum_1 = \begin{bmatrix} 0.04 & 0 \\ 0 & 4 \end{bmatrix}$

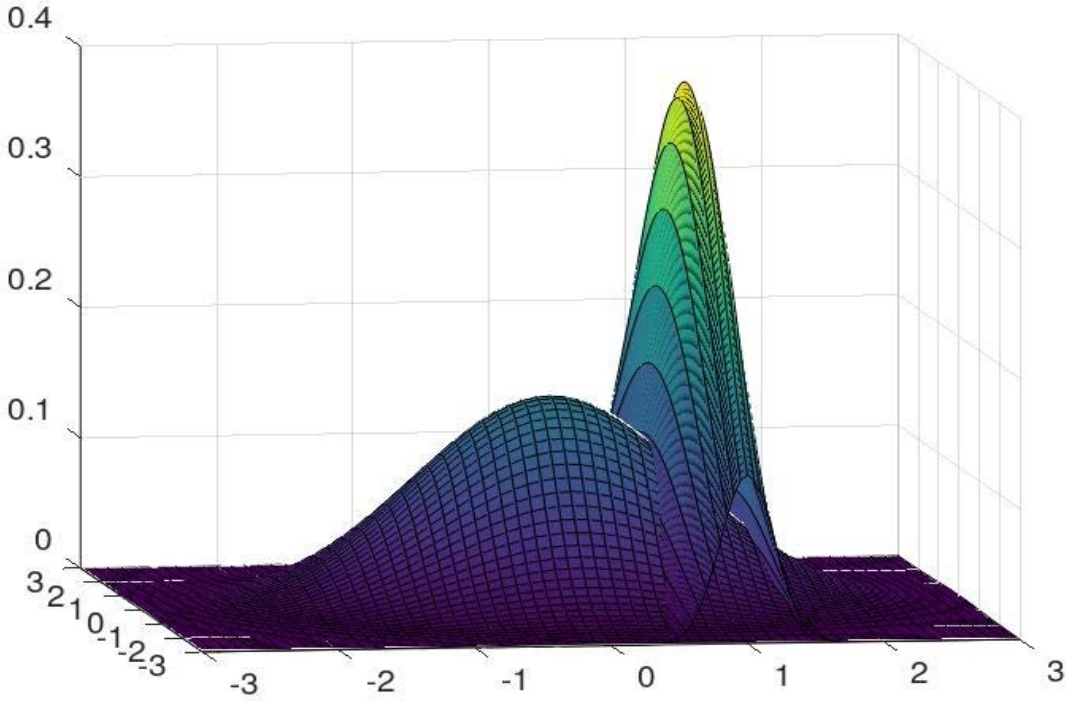
p has mean $\mu_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ and variance $\sum_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

(a) Compute the Entropy $H(p)$ and $H(q)$. Which one is larger? Why?

$H(p) = \frac{1}{2}log|\sum_2| + 1 + log(2\pi) = 1 + log(2\pi)$

$H(q) = \frac{1}{2}log|\sum_1| + 1 + log(2\pi) = log(0.4) + 1 + log(2\pi)$

When compared to p, q has been compressed by a factor of 5 in one direction but stretched by a factor of 2 in the other, resulting an overall



compression factor of 0.4. Consequently, $H(q)$ is smaller than $H(p)$ because the probability distribution is more concentrated and therefore less "uncertain". $H(p)$ is larger than $H(q)$ by $-log(0.4) = log(2.5)$

(b) Compute the KL-Divergence in each direction $D_{KL}(q||p)$ and $D_{KL}(p||q)$. Which one is larger? Why?

$D_{KL}(q \parallel p) = \frac{1}{2}\big[\parallel \mu \parallel^2 + \text{Trace}(\sum_1) - log|\sum_1| - d\big]/2$

$= [1 + 4.04 - log(0.16) - 2]/2 = 1.52 - log(0.4)$

$D_{KL}(p \parallel q) = \big[(\mu_1 - \mu_2)^T \sum_1^{-1} (\mu_1 - \mu_2) + \text{Trace}(\sum_1^{-1} \sum_2) + log|\sum_1| - log|\sum_2| - d\big]/2$

$= [25 + 25.25 + log(0.16) - log(1) - 2]/2$

$= 24.125 + log(0.4)$

The second one is much larger, because there are places where p is large but q is very close to zero.

(c) Compute the Wasserstein Distance $W_2(q, p)$

$W_2(q,p)^2 = \parallel \mu_1 - \mu_2 \parallel^2 + \text{Trace}(\sum_1 + \sum_2 - 2(\sum_1 \sum_2)^{\frac{1}{2}})$

$= 1 + 4.04 + 2 - 2(2.2) = 1 + 1 + 0.64$

$= 2.64$

$$W_2(q,p)^2 = \sqrt{2.64}$$

4. **Any Other Questions**

   Any further questions or discussion about PyTorch, other parts of the course, or broader implications of deep learning.