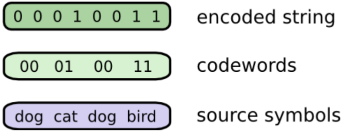
Introduction to Machine Learning

Entropy and Optimal Code Length



0 0 0 1 0 0 1 1 encoded string

00 01 00 11 codewords

dog cat dog bird source symbols

Learning goals

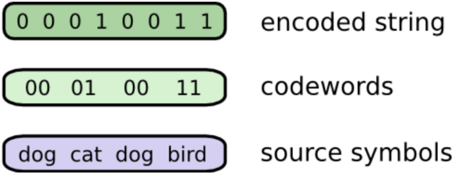
- Know that source coding is about encoding messages efficiently
- Know how to compute the average length of a code
- Know that the entropy of the source distribution is the lower bound for the average code length

SOURCE CODING

- There is an interesting connection between entropy and a subfield of information theory known as **source coding**.
- Abstractly, a source is any system or process that generates messages or information.
- A code is simply a way to represent the message so that it can be stored or transmitted over a communication channel (such as radio or fiber-optic cables).
- For example, one could use binary strings (0's and 1's) to encode messages.
- Because it may be expensive to transmit or store information, an important problem addressed by source coding is efficient coding schemes of minimal average length.

SOURCE CODING

- Formally, given a discrete alphabet/dictionary X of message symbols, a **binary code** is a mapping from symbols in X to a set of codewords of binary strings.
- For example, if our dictionary only consists of the words "dog", "cat", "fish" and "bird", each word can be encoded as a binary string of length 2 : "dog" → **00**, "cat" → **01**, "fish" → **10** and "bird" → **11**.
- For this code, a binary string can be decoded by replacing each successive pair of digits with the associated word.

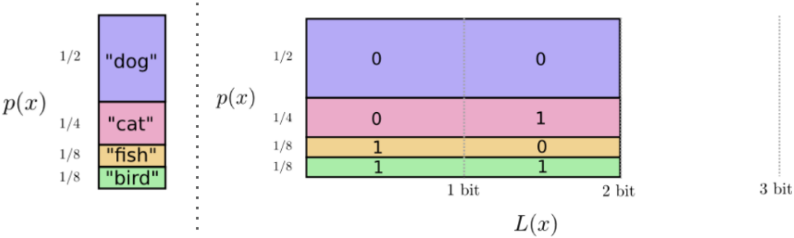


Credit: Chris Olah

Chris Olah (2015): Visual Information Theory. http://colah.github.io/posts/2015-09-Visual-Information/

SOURCE CODING

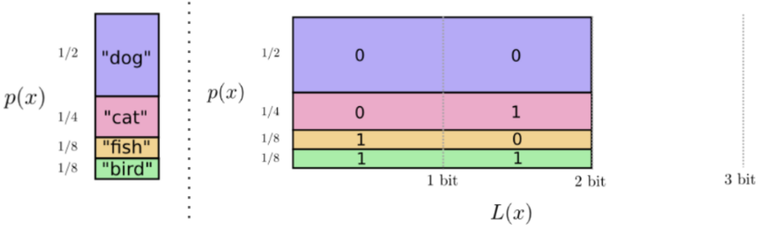
- Encoded messages are emitted by a source which can be modeled as a probability distribution over the message symbols in the dictionary.
- Let X be a random variable that represents a symbol from our data source and let $p(x) = \mathbb{P}(X = x)$, for symbol x in our dictionary.



Credit: Chris Olah

- The length $L(x)$ is simply the number of bits in the corresponding codeword. In this example, all codewords have length 2.

SOURCE CODING



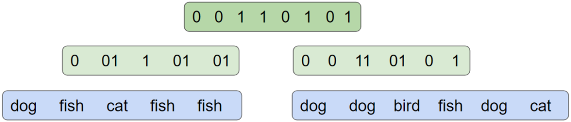
- For this code, the expected length of a message emitted by the source is, naturally:

$$\mathbb{E}[L(X)] = \frac{1}{2} \cdot 2 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 2 + \frac{1}{8} \cdot 2 = 2 \text{ bits.}$$

- The area of a rectangle in the image on the right reflects the size of the corresponding term in the expectation.

SOURCE CODING

- Maybe we can create better average-length coding schemes with **variable-length** codes by assigning shorter codes to more likely messages and longer one to less likely messages.
- However, this can be problematic because we want the receiver to be able to unambiguously decode the encoded string.
- Let us say the words in our dictionary are encoded in this way: "dog" → **0**, "cat" → **1**, "fish" → **01** and "bird" → **11**.
- In this case, the string 00110101 can be decoded in multiple ways.



0 0 1 1 0 1 0 1

0 01 1 01 01

dog fish cat fish fish

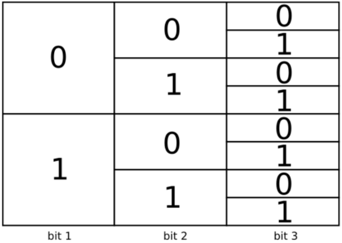
0 0 11 01 0 1

dog dog bird fish dog cat

- One way to make variable-length messages unambiguous is by ensuring that no codeword is a prefix (initial segment) of any other codeword. Such a code is known as a **prefix code**.

SOURCE CODING

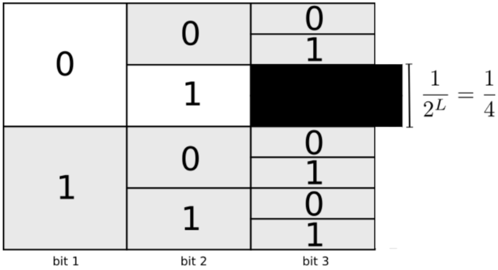
- In general, the number of possible codewords grows exponentially in length L.
- For binary codes, there are two possible words of length one, four possible words of length two and 2^L possible words of length L.



bit 1	bit 2	bit 3
0	0	0
		1
	1	0
		1
1	0	0
		1
	1	0
		1

- In total, there are $(2^{L+1} - 2)$ codewords of length $\leq$ L.

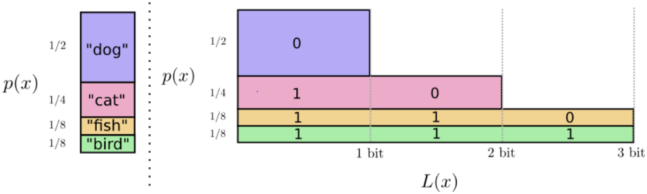
SOURCE CODING



- Here, if the codeword **01** is assigned to a symbol, then **010** and **011** cannot be assigned to any other symbol because that would break the prefix property.
- If a codeword of length L is assigned to a symbol, then $\frac{1}{2^L}$ of the possible codewords of length $> L$ must be discarded.
- If some symbols are assigned short codewords, due to the prefix property, many marginally longer codewords cannot be assigned to other symbols.

SOURCE CODING

- An example of prefix code:
  "dog" → **0**, "cat" → **10**, "fish" → **110** and "bird" → **111**.



- Here, the expected code length is :

$$
\begin{aligned}
\mathbb{E}[L(X)] &= \frac{1}{2}\cdot 1 + \frac{1}{4}\cdot 2 + \frac{1}{8}\cdot 3 + \frac{1}{8}\cdot 3 \\
&= -\frac{1}{2}\cdot \log_2\left(\frac{1}{2}\right) - \frac{1}{4}\cdot \log_2\left(\frac{1}{4}\right) - \frac{1}{8}\cdot \log_2\left(\frac{1}{8}\right) - \frac{1}{8}\cdot \log_2\left(\frac{1}{8}\right) \\
&= H(X) = 1.75 \text{ bits. } (< 2 \text{ bits})
\end{aligned}
$$

- Actually, this coding scheme is the most efficient way to store and transmit these messages. It is simply not possible to do better!

SOURCE CODING

- In fact, Shannon's **source coding theorem** (or **noiseless coding theorem**) tells us that the optimal trade-off is made when the code length of a symbol with probability p is $\log(1/p)$.
- In other words, the entropy of the source distribution is the theoretical lower bound on the average code length.
- If it is any lower, some information will be distorted or lost.
- In practice, algorithms such as Huffman Coding can be used to find variable-length codes that are close (in terms of expected length) to the theoretical limit.