AIMS | African Institute for Mathematical Sciences RWANDA

AMMI Review sessions

Deep Learning (7)

Concepts from Information Theory

2nd March 2020

Concepts from Information Theory

Information Theory

- Information theory is a branch of applied mathematics that revolves around quantifying how much information is present in a signal

- It was originally invented to study sending messages from discrete alphabets over a noisy channel, such as communication via radio transmission.

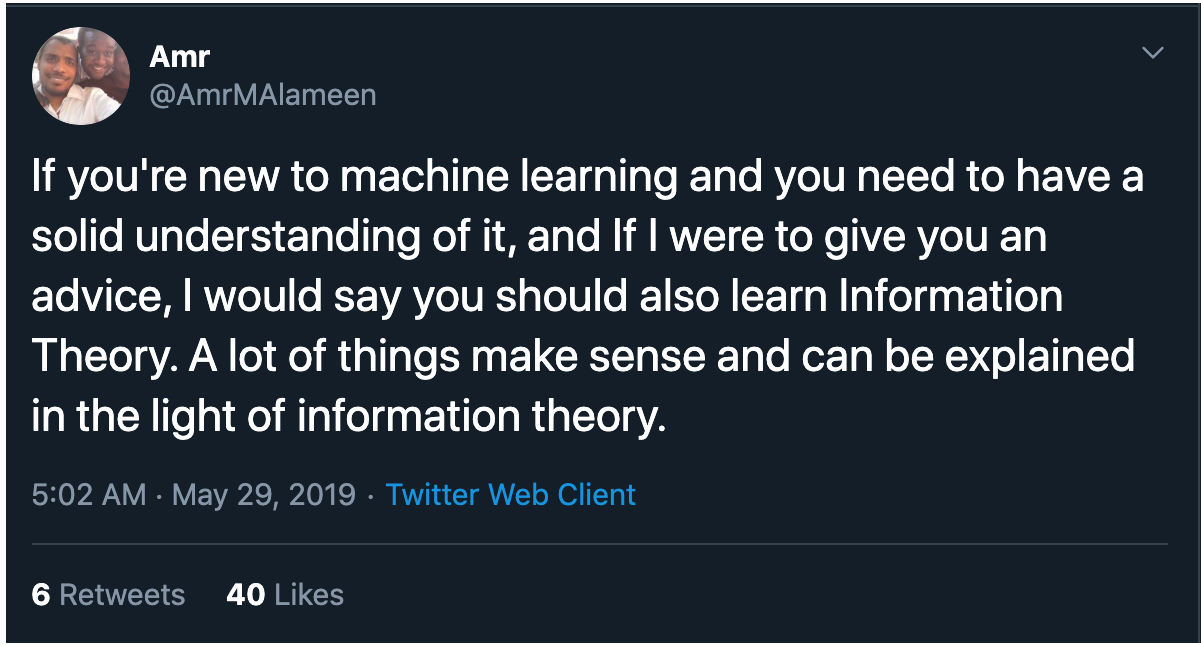
- In this context, information theory tells how to design optimal codes and calculate the expected length of messages sampled from specific probability distributions using various encoding schemes

Concepts from Information Theory

- Some examples of concepts in AI that come from Information theory or related fields:

- Popular cross-entropy loss function
- Building decision trees on basis of maximum information gain
- Viterbi algorithm widely used in NLP and Speech
- Concept of encoder-decoder popularly used in

Machine Translation RNNs and various other type of models



Entropy

- The basic intuition behind information theory is that learning that an unlikely event has occurred is more informative than learning that a likely event has occurred.

- A message saying “the sun rose this morning” is so uninformative as to be unnecessary to send, but a message saying “there was a solar eclipse this morning” is very informative.

Entropy

We would like to quantify information in a way that formalizes this intuition. Specifically,

- Likely events should have low information content, and in the extreme case, events that are guaranteed to happen should have no information content whatsoever.
- Less likely events should have higher information content.
- Independent events should have additive information. For example, finding out that a tossed coin has come up as heads twice should convey twice as much information as finding out that a tossed coin has come up as heads once.
- Other notations use base-2 logarithms and units called bits or **shannons**; information measured in bits is just a rescaling of information measured in nats.

Entropy

- We can quantify the amount of uncertainty in an entire probability distribution using the Shannon entropy:

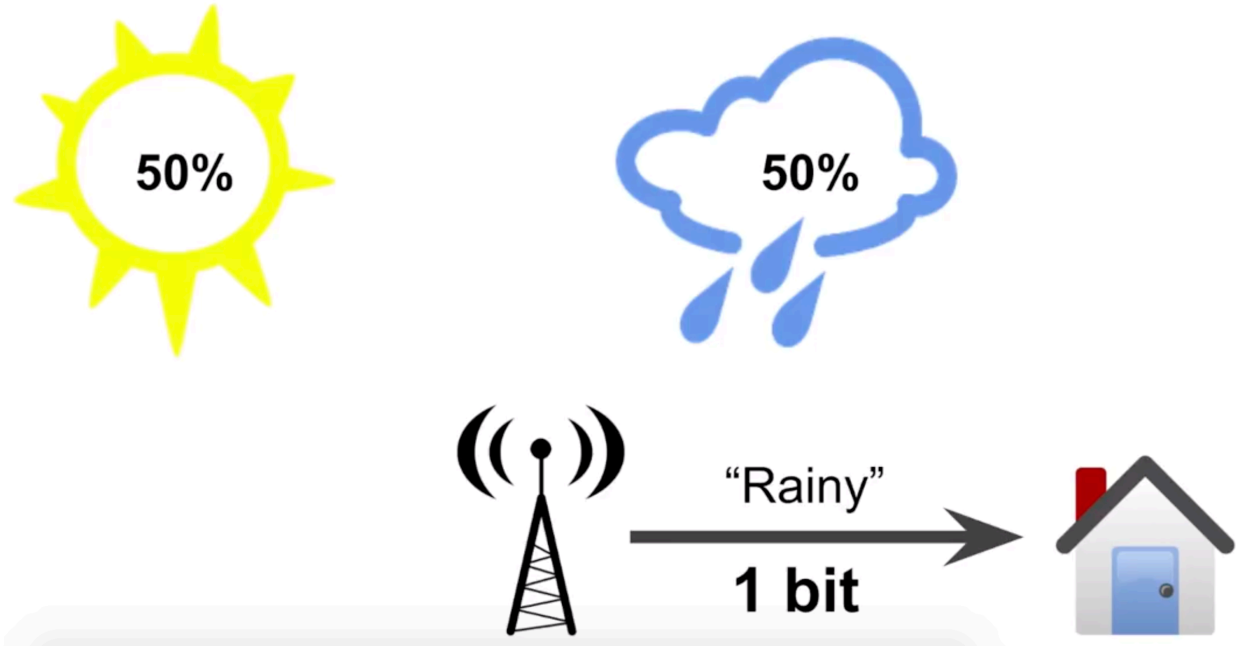
$$H(\mathrm{x}) = \mathbb{E}_{\mathrm{x}\sim P}[I(x)] = -\mathbb{E}_{\mathrm{x}\sim P}[\log P(x)].$$

- In other words, the Shannon entropy of a distribution H(P) is the expected amount of information in an event drawn from that distribution. It gives a lower bound on the number of bits (if the logarithm is base 2, otherwise the units are different) needed on average to encode symbols drawn from a distribution P.

Entropy Example

- If you communicate 1 bit of information, then you divided the uncertainty by factor of 2.

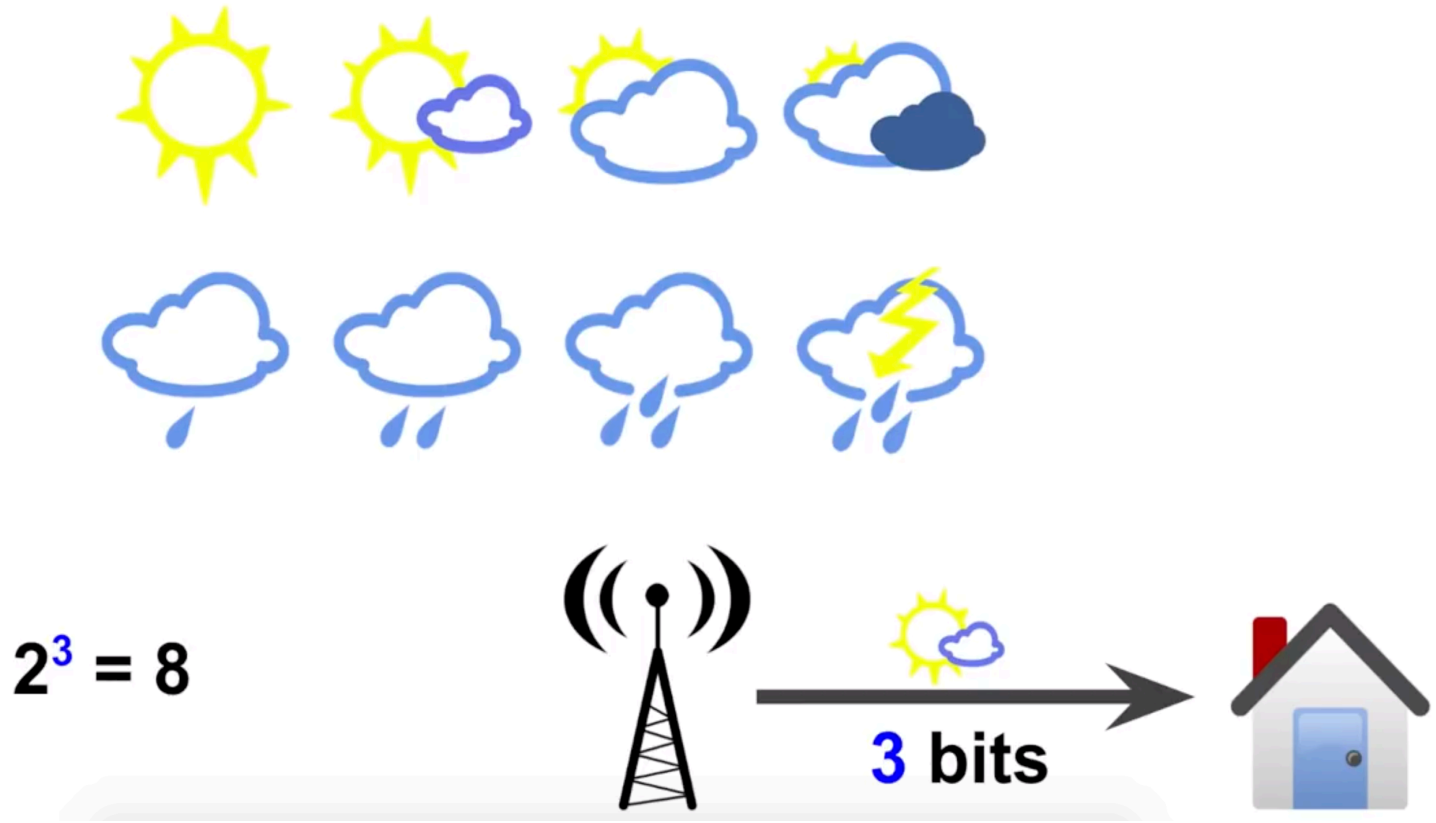
- No matter how you encode the message (for example in 40 bits string “Rainy”) it still have the same amount of useful information



Source: [Aurélien Géron, Introduction to Entropy, Cross-Entropy and KL-Divergence]

Entropy Example

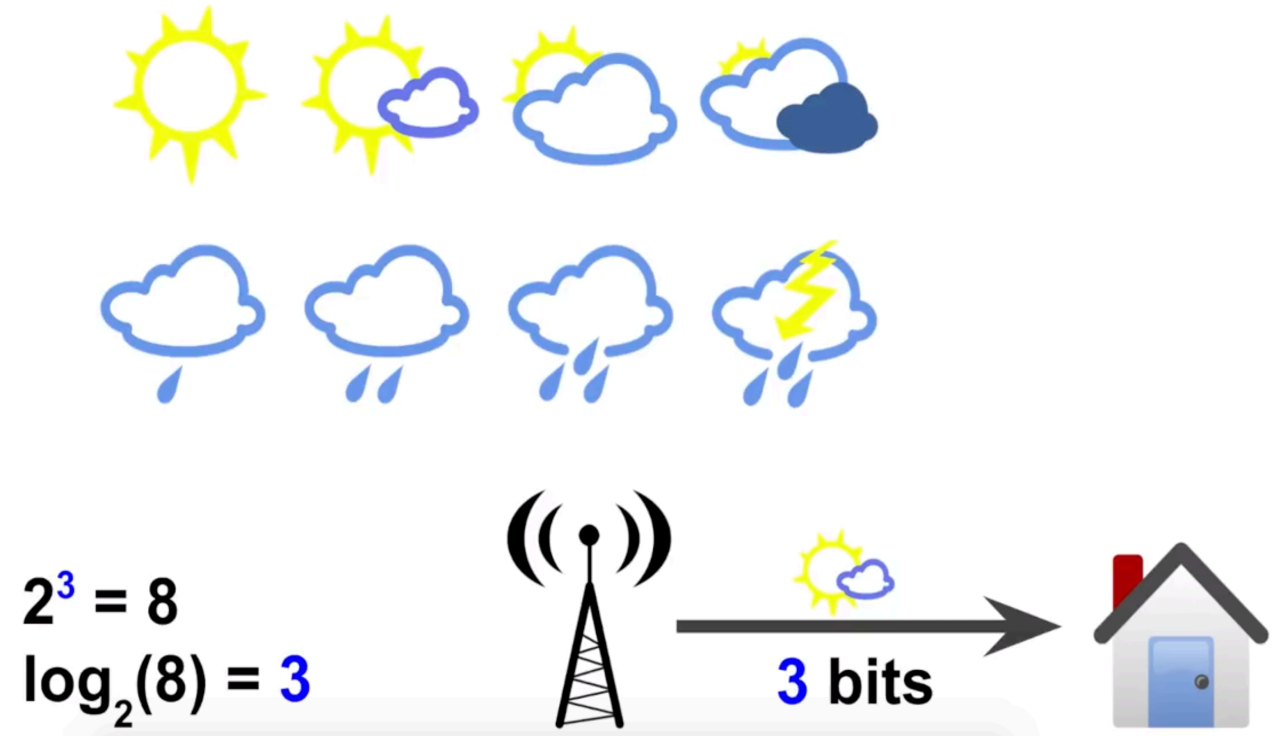
- if you have 8 equally likely events then you would use 3 bits to communicate the weather state



Entropy Example

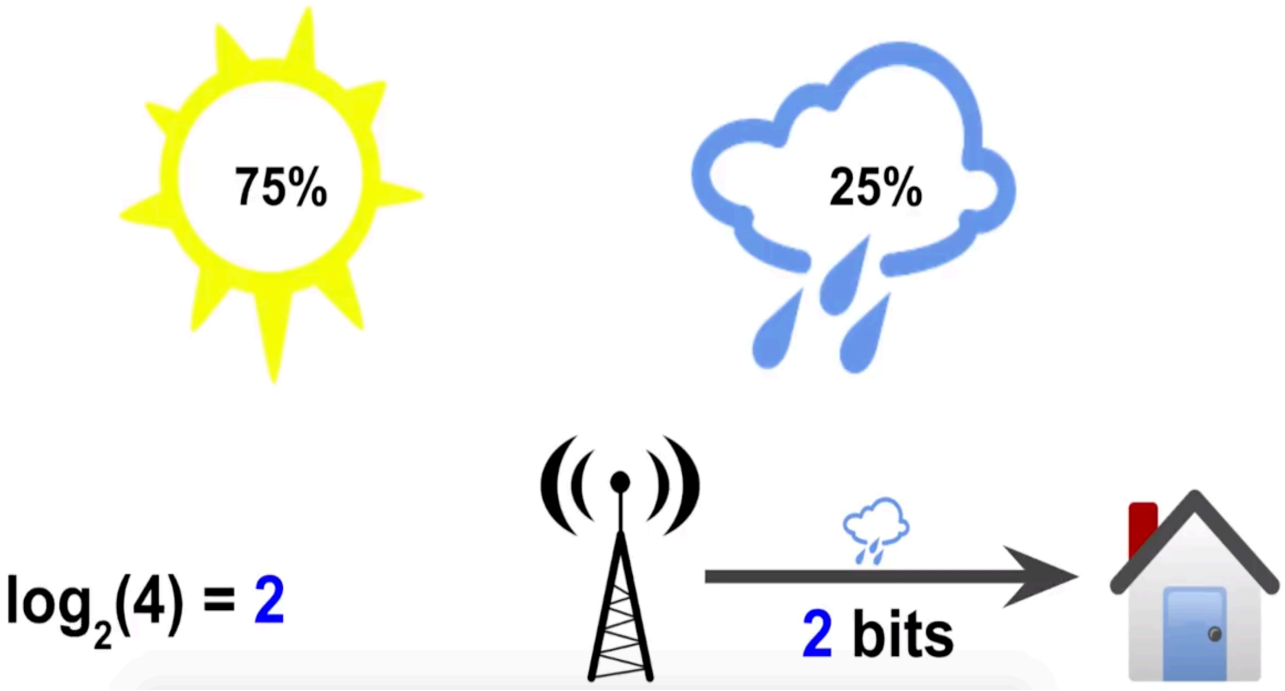
- We can find the number of bits by calculating the binary logarithm of the uncertainty reduction factor

Which equal to 8 in this example



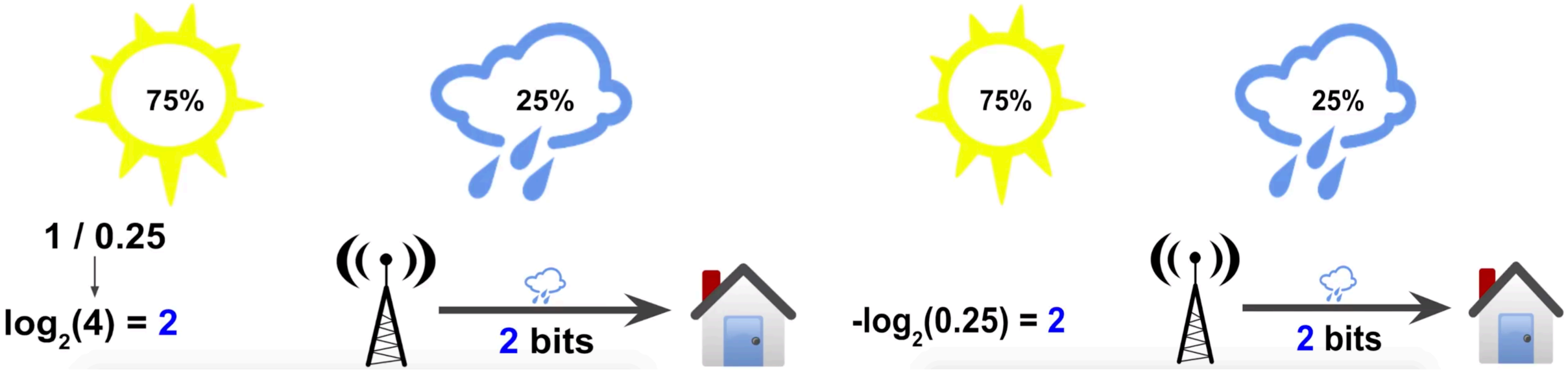
Entropy Example

- What If the possibilities are not equally likely?
- Communicating a rainy state in the following example will reduce the uncertainty by factor of 4



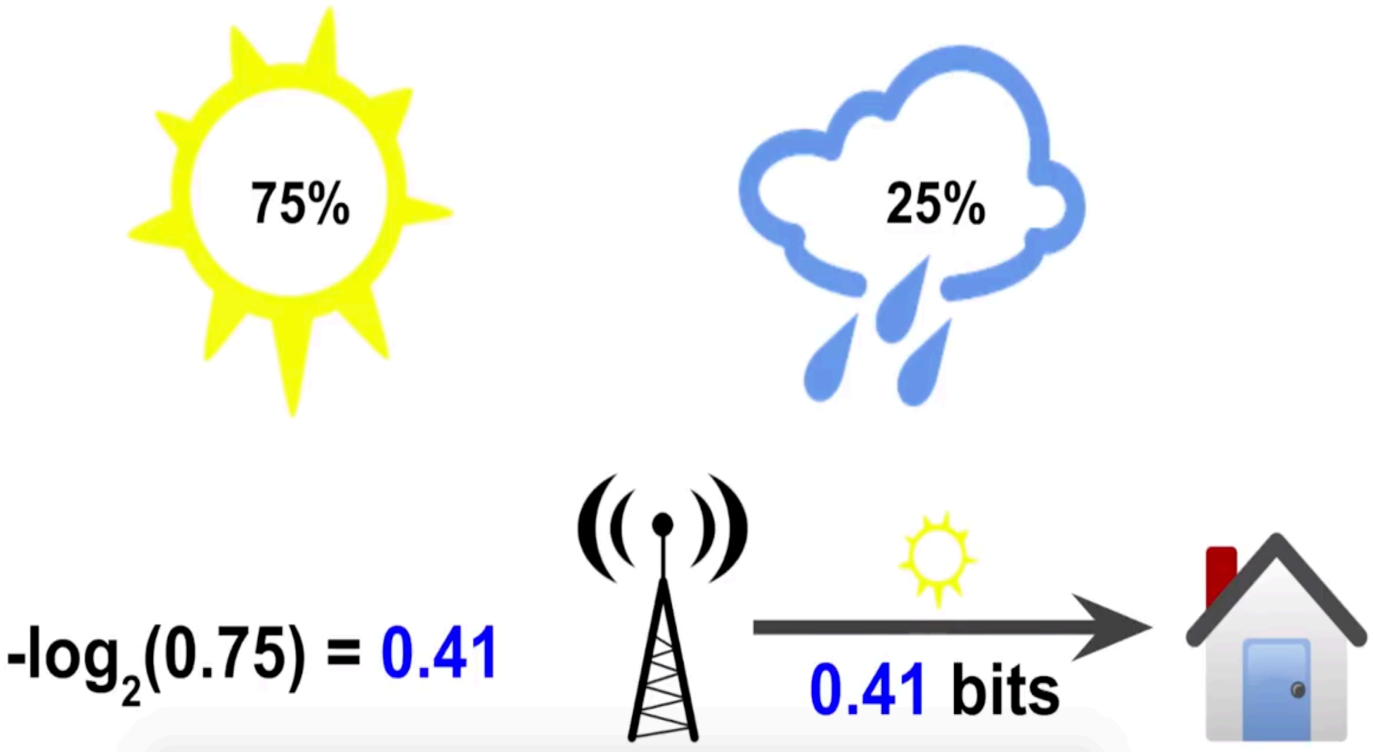
Entropy Example

- What If the possibilities are not equally likely?
- Communicating a rainy state in the following example will reduce the uncertainty by factor of 4
- It can be obtained also by calculating the – log the probability of event



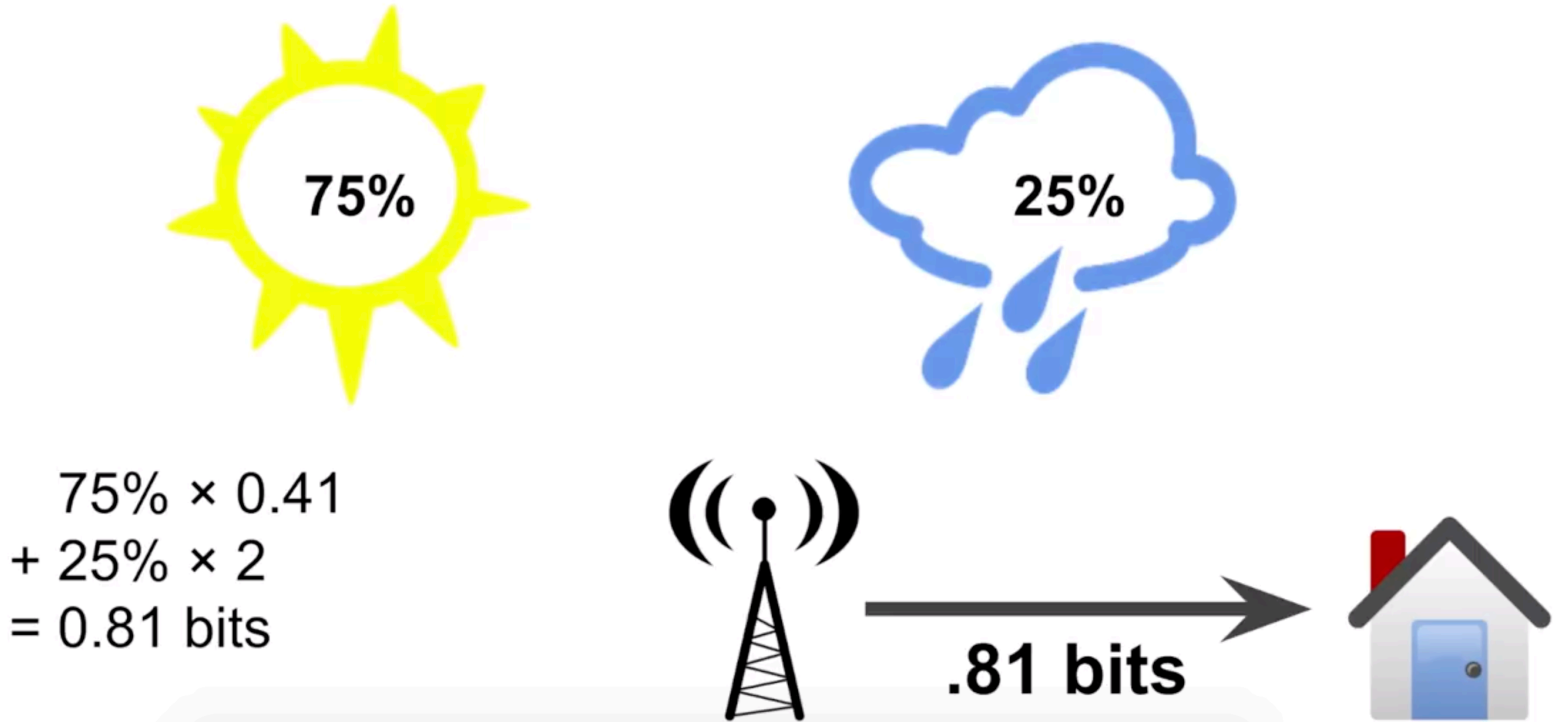
Entropy Example

- Same thing if the weather station will till you it will be sunny tomorrow, -log(0.75)
- As expected less number of bits needed since it's the most likely event!



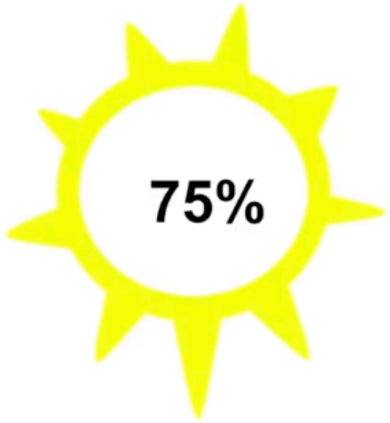
Entropy Example

- What Is the expected number of useful information (bits) you get from the weather station on average ?



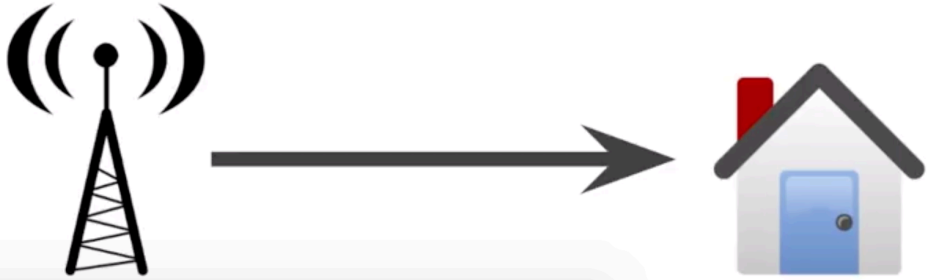
Entropy Example

- What we have calculated is the Entropy!
- A measure of how uncertain the events are, or the avg amount of information obtained from one sample drawn from a probability distribution, or how unpredictable that probability distribution is



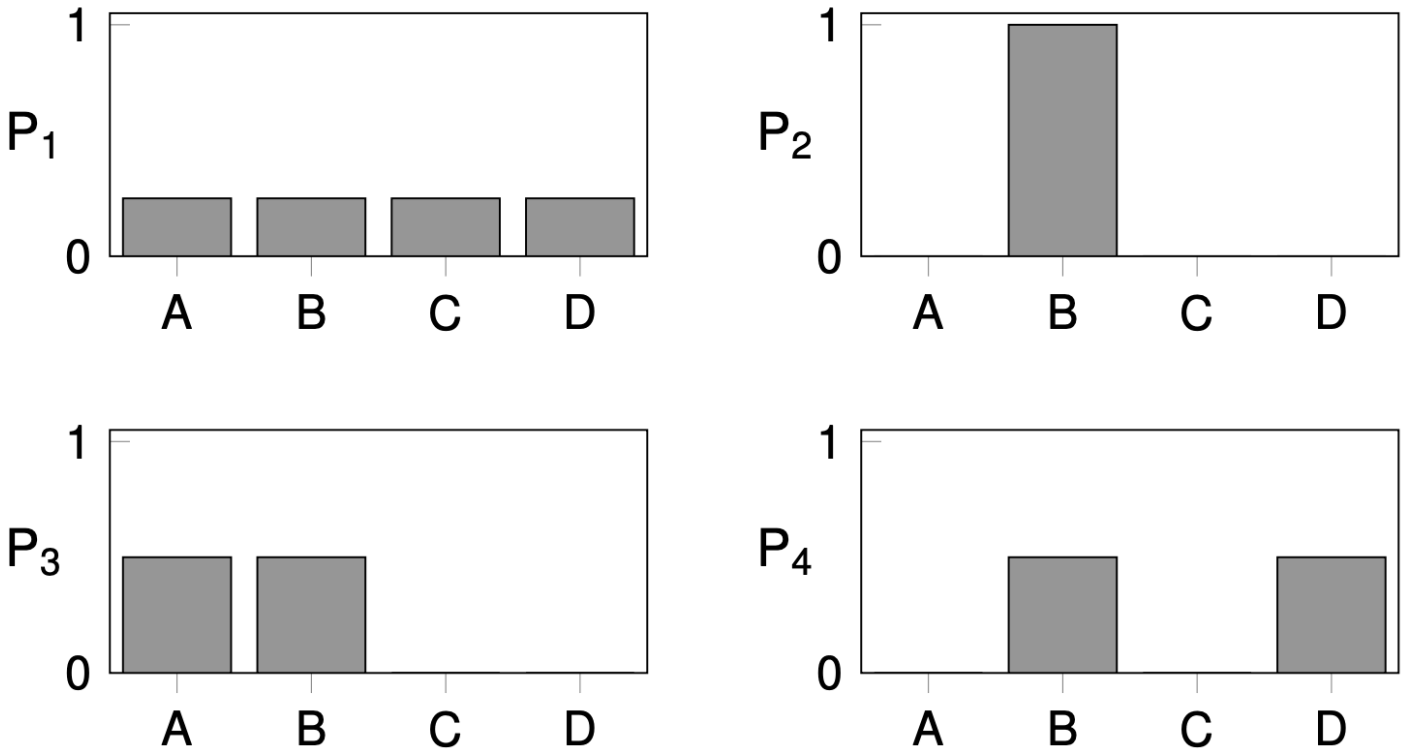
Entropy:

$$H(\mathbf{p}) = -\Sigma_i\, p_i \log_2(p_i)$$



Entropy (exercise)

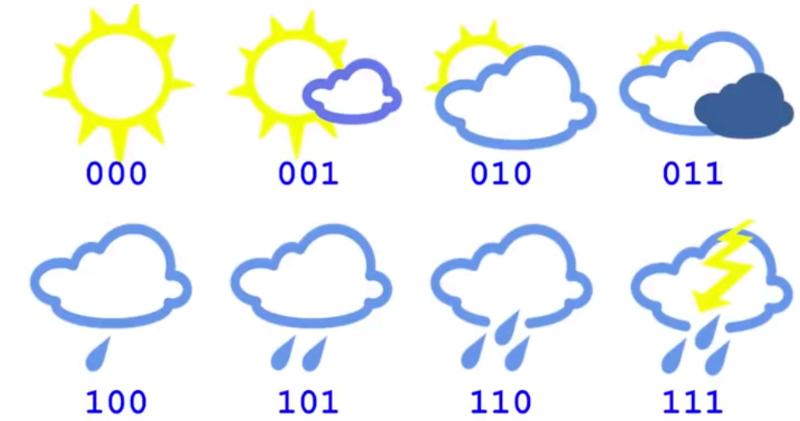
- Rank these distributions from highest to lowest entropy.



Uniform distributions have maximum uncertainty

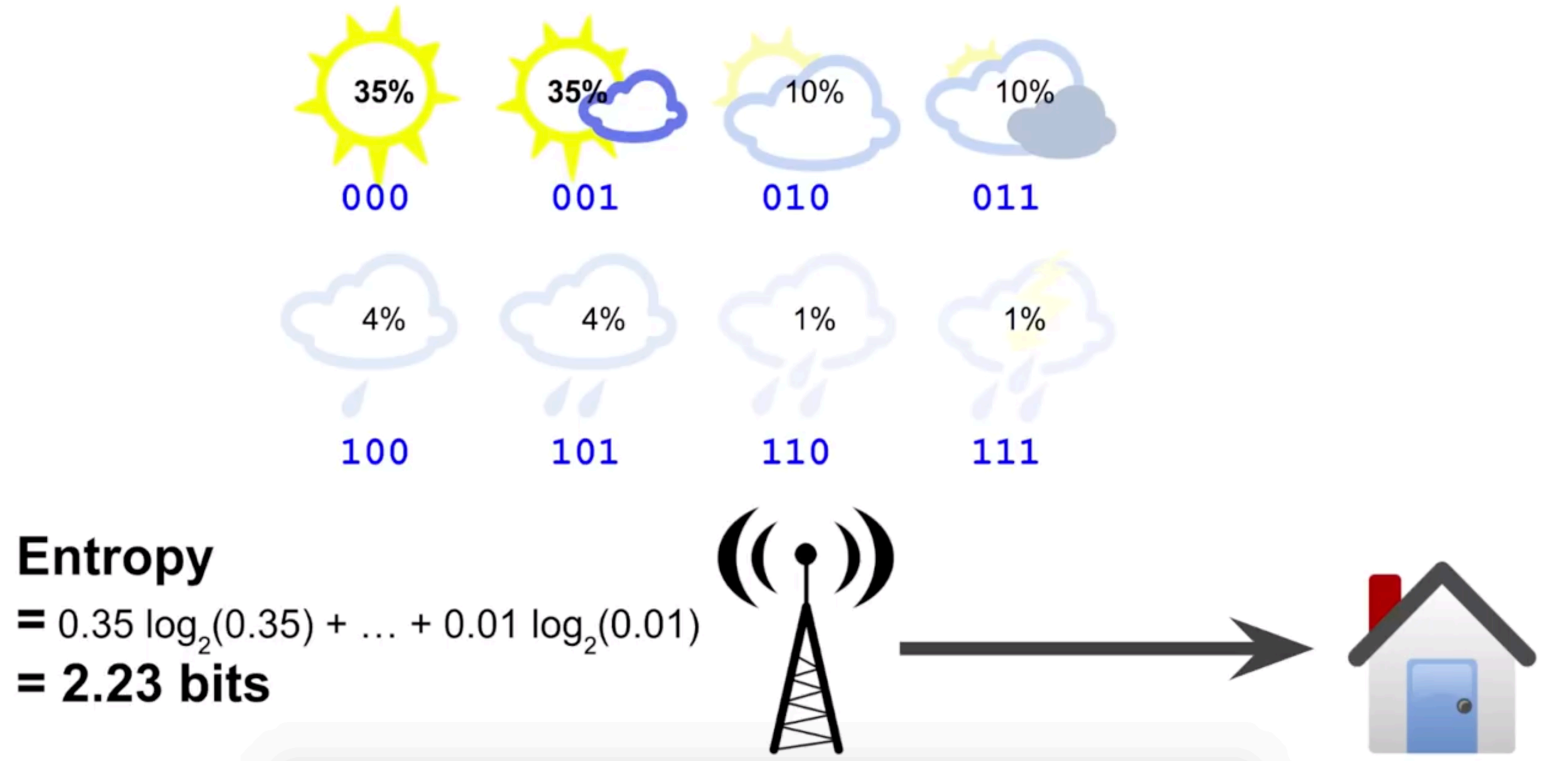
Cross Entropy Example

- Cross Entropy average message length
- Best scenario is when
  the average message length = the average amount of information per message (the entropy)
- For example if we encode every possible option using 3 bits the cross entropy = 3



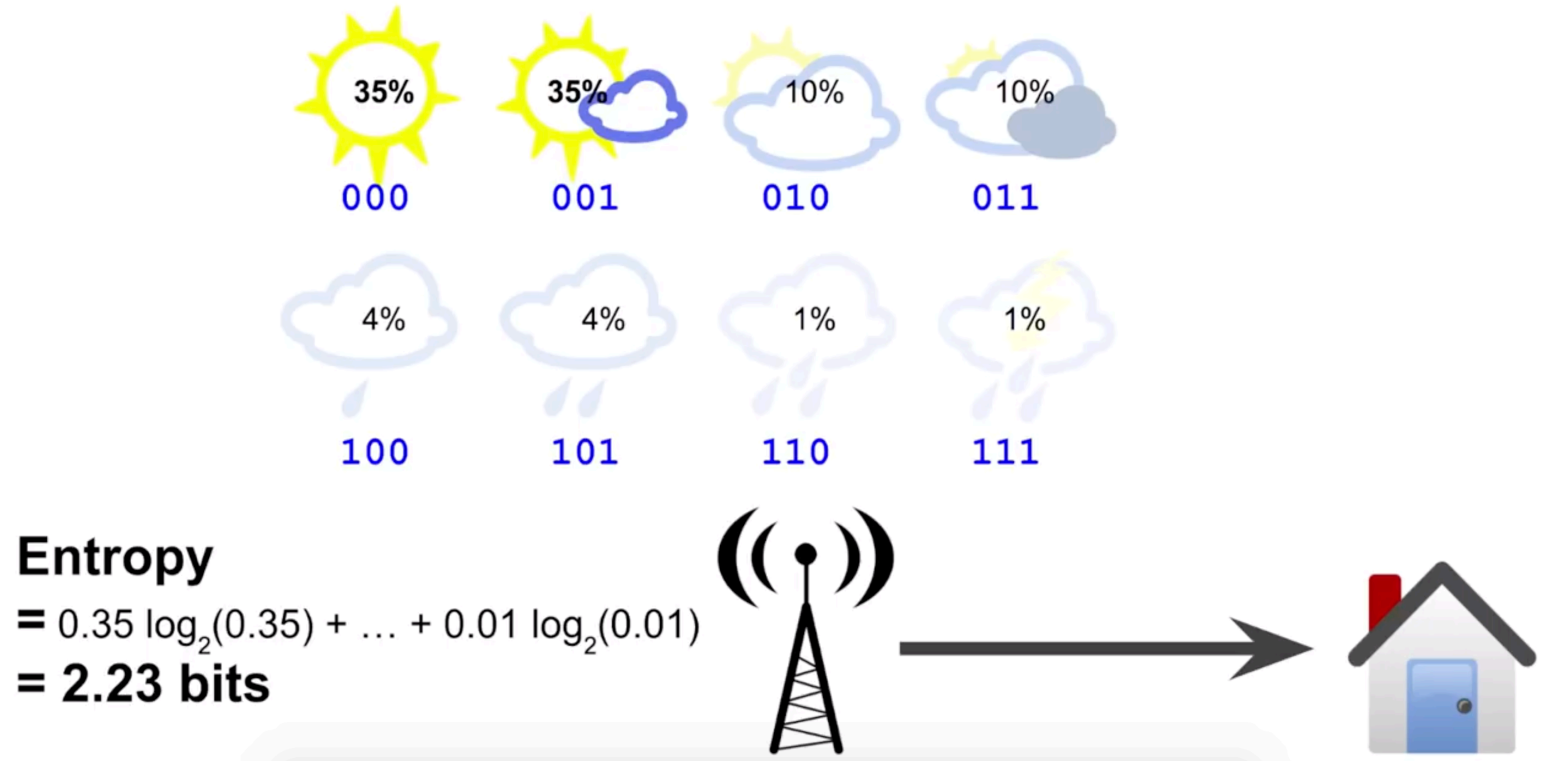
Cross Entropy Example

- Let’s assume for the previous example the probability distribution looks like the following
- By calculating the entropy we see that the entropy is less than 3, i.e 3 bits is not the optimal coding choice



Cross Entropy Example

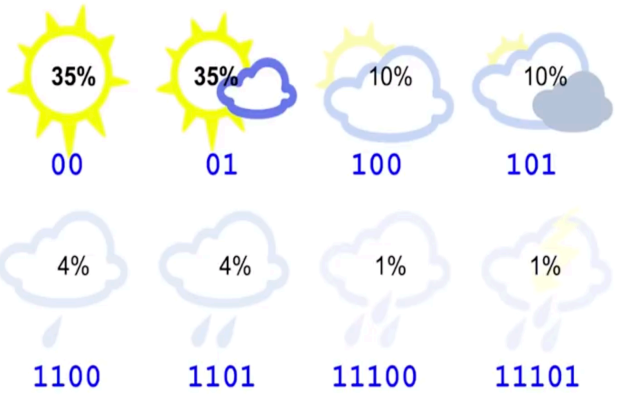
- Let’s assume for the previous example the probability distribution looks like the following
- By calculating the entropy we see that the entropy is less than 3, i.e 3 bits is not the optimal coding choice



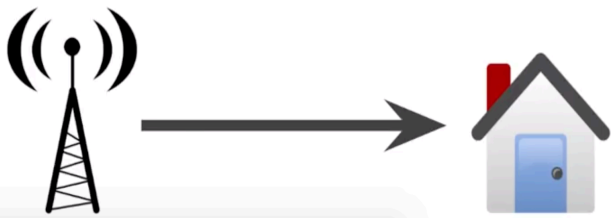
Cross Entropy Example

- Let’s assume we have changed the coding scheme to the two following options

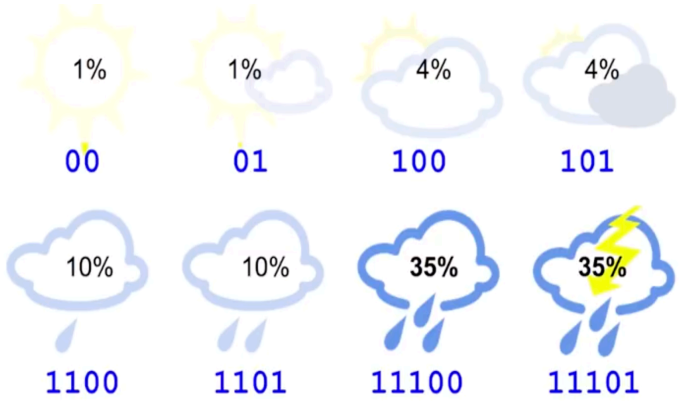
option (1) better than 3 bit coding scheme



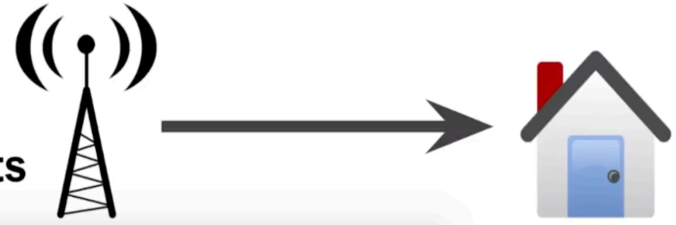
35%×2 + 35%×2 + 10%×3
+ 10%×3 + 4%×4 + 4%×4
+ 1%×5 + 1%×5 = **2.42 bits**



option (2) worse than 3 bit coding scheme

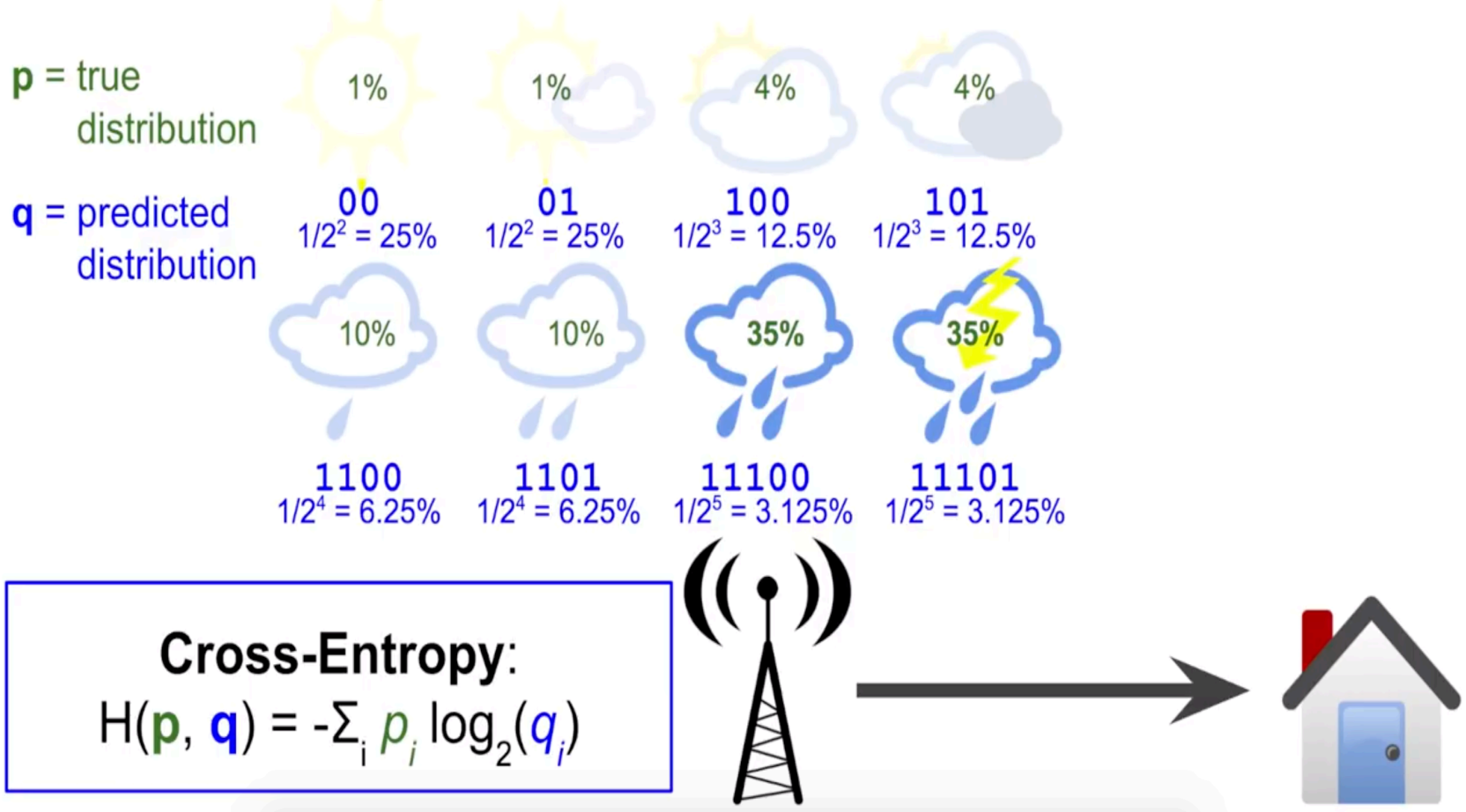


1%×2 + 1%×2 + 4%×3 +
4%×3 + 10%×4 + 10%×4
+ 35%×5 + 35%×5 = **4.58 bits**



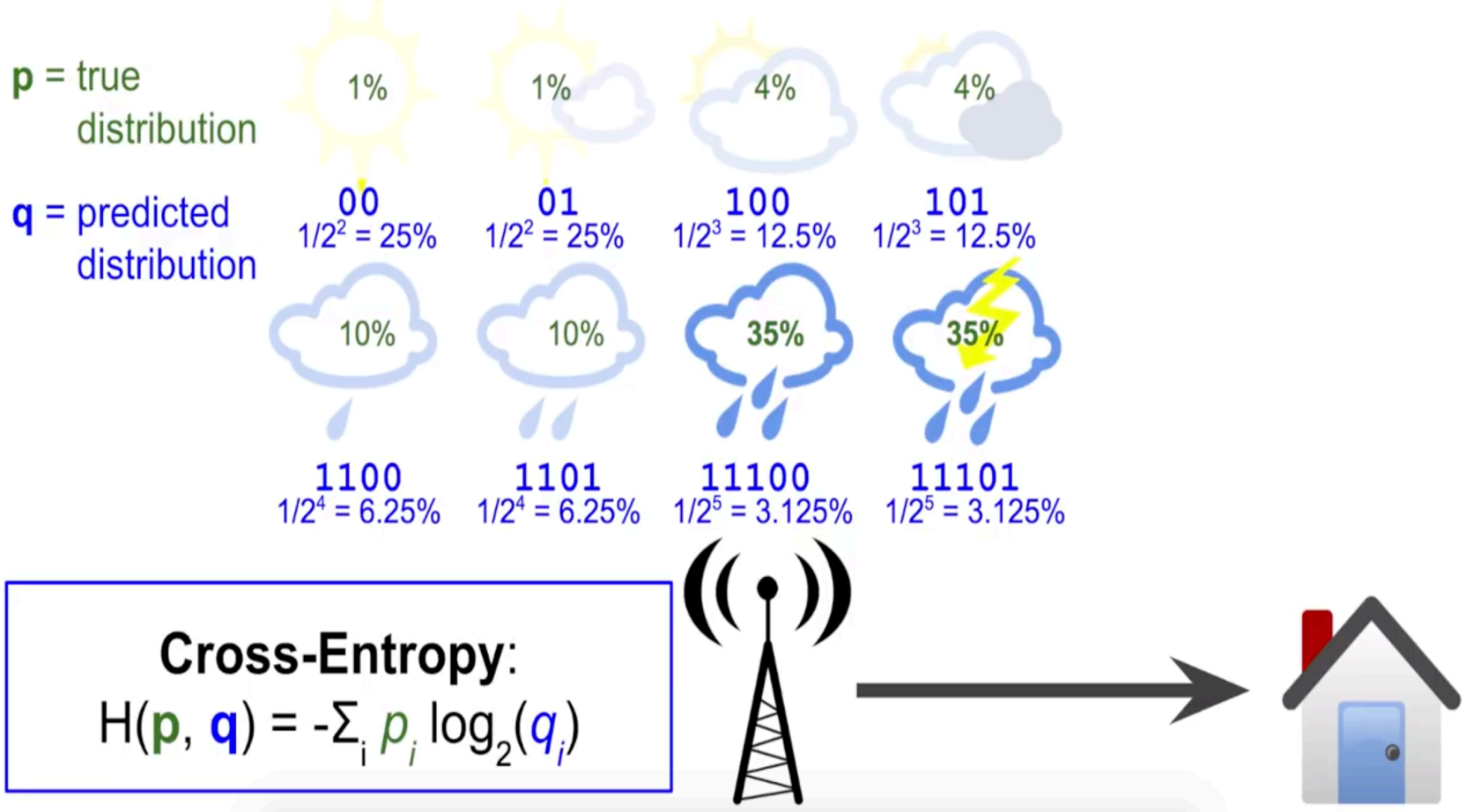
Cross Entropy

- The codes we are using make implicit assumptions about the weather distribution
- for example the 2 bits message for the sunny weather implicitly assumes will have a sunny day every four days



Cross Entropy

- The codes we are using make implicit assumptions about the weather distribution
- for example the 2 bits message for the sunny weather implicitly assumes will have a sunny day every four days



Kullback–Leibler divergence

- If our predicted distribution equal to the true distribution, then Cross entropy = entropy
- Otherwise the cross entropy will be greater than the entropy by some number of bits, this is amount is called the **Kullback–Leibler divergence, KL divergence or relative entropy.**
- Cross Entropy = Entropy + KL divergence

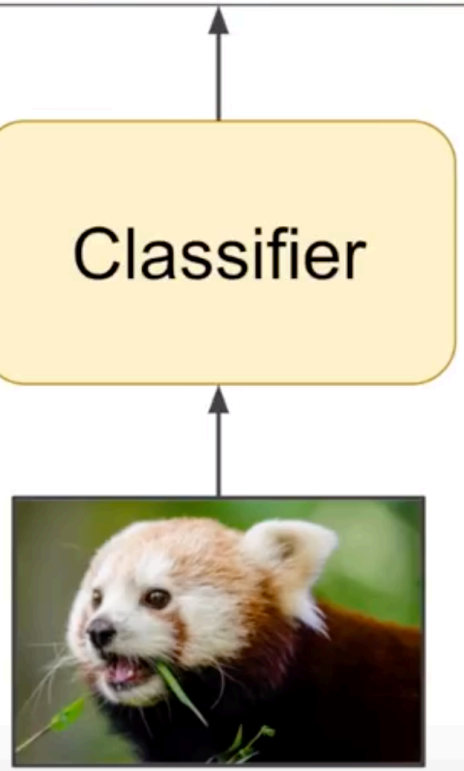
KL Divergence:
$D_{KL}(p \,||\, q) = H(p, q) - H(p)$

- In the previous example:

KL Divergence:
$D_{KL}(p \,||\, q) = 4.58 - 2.23$

Cross Entropy as a cost function

	Cat	Dog	Fox	Cow	Red Panda	Bear	Dolphin
True distribution:	0%	0%	0%	0%	100%	0%	0%
Predicted distribution:	2%	30%	45%	0%	25%	5%	0%



Cross-Entropy Loss:

$H(\mathbf{p}, \mathbf{q}) = -\Sigma_i \, p_i \log(q_i)$

$= -\log(0.25) = 1.386$

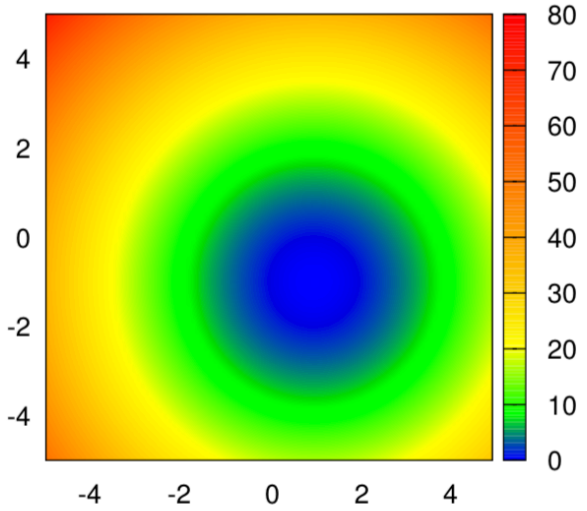
$\log_2(x) = \log(x) / \log(2)$

Cross Entropy vs MSE for classification

- In practice MSE penalizes responses "too strongly on the right side".

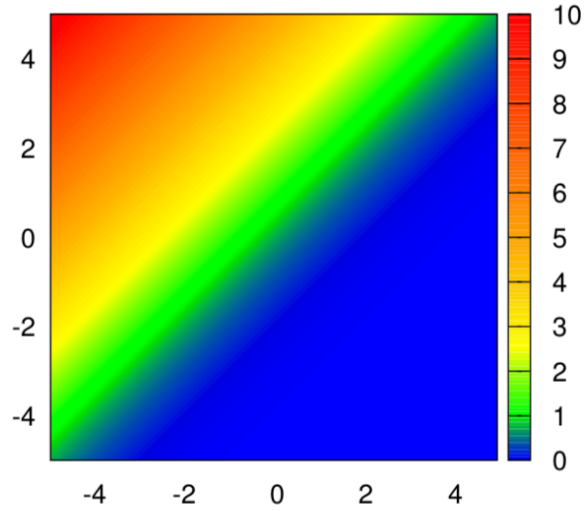
Let's consider the loss for a single sample in a two-class problem, with a predictor with two output values. The x axis here is the activation of the correct output unit, and the y axis is the activation of the other one.

MSE



$$\mathscr{L} = (x-1)^2 + (y+1)^2$$

Cross-entropy



$$\mathscr{L} = -\log\left(\frac{e^x}{e^x+e^y}\right)$$

MSE incorrectly penalizes outputs which are perfectly valid for prediction, contrary to cross-entropy.

Cross Entropy in Pytorch

This is precisely the value of `torch.nn.CrossEntropyLoss`.

```
>>> f = torch.tensor([[-1., -3., 4.], [-3., 3., -1.]])
>>> target = torch.tensor([0, 1])
>>> criterion = torch.nn.CrossEntropyLoss()
>>> criterion(f, target)
tensor(2.5141)
```

and indeed

$$-\frac{1}{2}\left(\log\frac{e^{-1}}{e^{-1}+e^{-3}+e^{4}}+\log\frac{e^{3}}{e^{-3}+e^{3}+e^{-1}}\right)\simeq 2.5141.$$

Cross Entropy in Pytorch

The cross-entropy loss can be seen as the composition of a "log soft-max" to normalize the score into logs of probabilities

$$(\alpha_1, \ldots, \alpha_C) \mapsto \left(\log \frac{\exp \alpha_1}{\sum_k \exp \alpha_k}, \ldots, \log \frac{\exp \alpha_C}{\sum_k \exp \alpha_k} \right),$$

which can be done with the `torch.nn.LogSoftmax` module, and a read-out of the normalized score of the correct class

$$\mathscr{L}(w) = -\frac{1}{N} \sum_{n=1}^{N} f_{y_n}(x; w),$$

which is implemented by the `torch.nn.NLLLoss` criterion.

Hence, if a network should compute log-probabilities, it may have a `torch.nn.LogSoftmax` final layer, and be trained with `torch.nn.NLLLoss`.

Statistical Aspects of KL divergence

- Let's define the Likelihood ration between two random variables as

$$LR = \frac{p(x)}{q(x)}$$

- For any sample **x** the ratio between the likelihoods indicates how much more likely the data-point is to occur in **p(x)** as opposed to **q(x)**. So, a value larger than 1 indicates that **p(x)** is the more likely model, whereas a value smaller than 1 indicates the opposite, **q(x)** is more likely.

- For a set of data with independent samples you can compute the likelihood ratio for the entire set by taking the product of the likelihood ratio for each sample.

$$LR = \prod_{i=0}^{n} \frac{p(x_i)}{q(x_i)}$$

Statistical Aspects of KL divergence

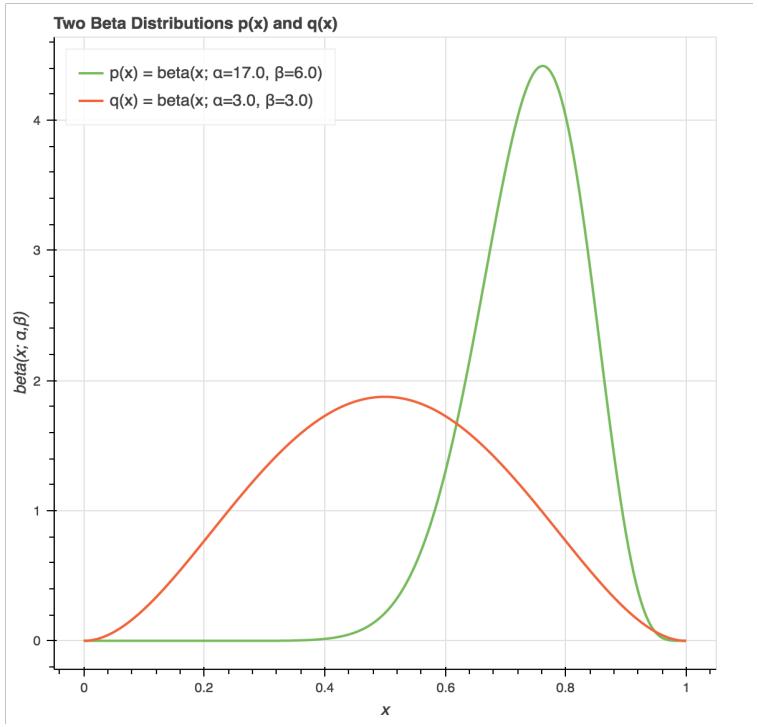
- If we take the log of the likelihood

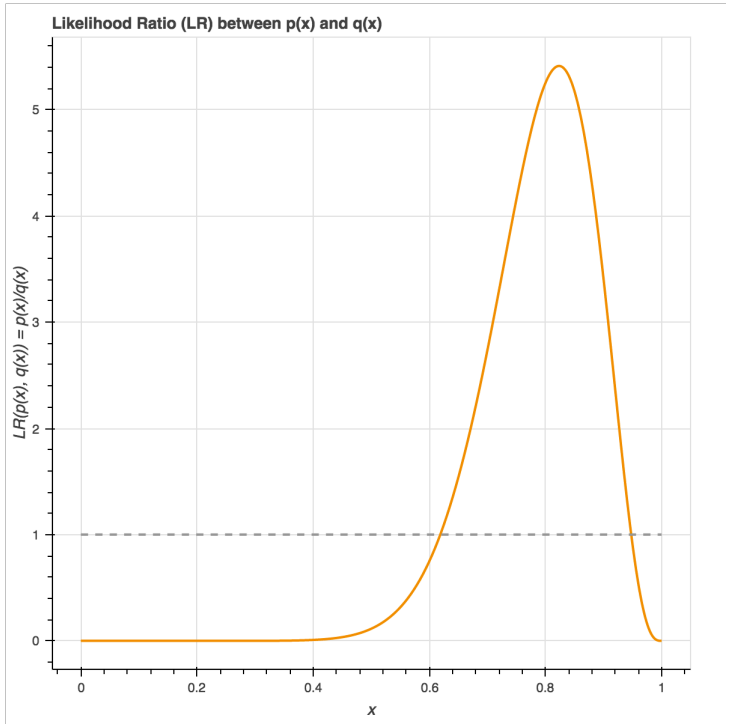
$$\log_{10} LR = \sum_{i=0}^{n} \log_{10} \left(\frac{p(x_i)}{q(x_i)} \right)$$

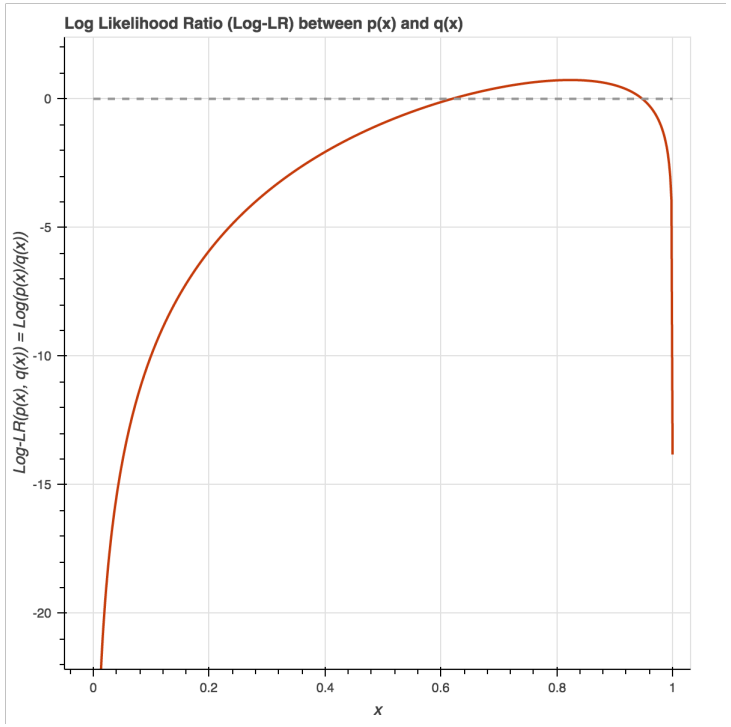
- Then a zero value indicate that **p(x)** better fits the data, whereas a value less than zero tells us that **q(x)** better fits the data, A value of zero indicates that both models fit the data equally well.

Statistical Aspects of KL divergence

- For example suppose **p(x)** and **q(x)** to be beta distributions with parameters (a=17, b= 6) for **p(x)** and (a=3, b=3) for **q(x)** respectively, then their ***LR*** and **log *LR*** curves look the following way:







Statistical Aspects of KL divergence

- What happens if you do this for an infinite amount of samples from **p(x)**? If you let **N→infinity**, then you get the following:

$$x_i \overset{i.i.d.}{\sim} p(x)$$

$$\lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N} \log \frac{p(x_i)}{q(x_i)} = \mathbb{E}_{x\sim p(x)} \left\{ \log \frac{p(x_i)}{q(x_i)} \right\}$$

- analytically is defined as

$$\mathbb{E}_{p(x)} \left\{ \log \left(\frac{p(x)}{q(x)} \right) \right\} = \int_{-\infty}^{\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) dx$$

- And this is the exact definition of the KL divergence!

$$D(p(x), q(x)) = \int_{-\infty}^{\infty} p(x) \log \left(\frac{p(x)}{q(x)} \right) dx$$

Statistical Aspects of KL divergence

- Intuitively how much will each sample on average till about the distinction between the two distributions

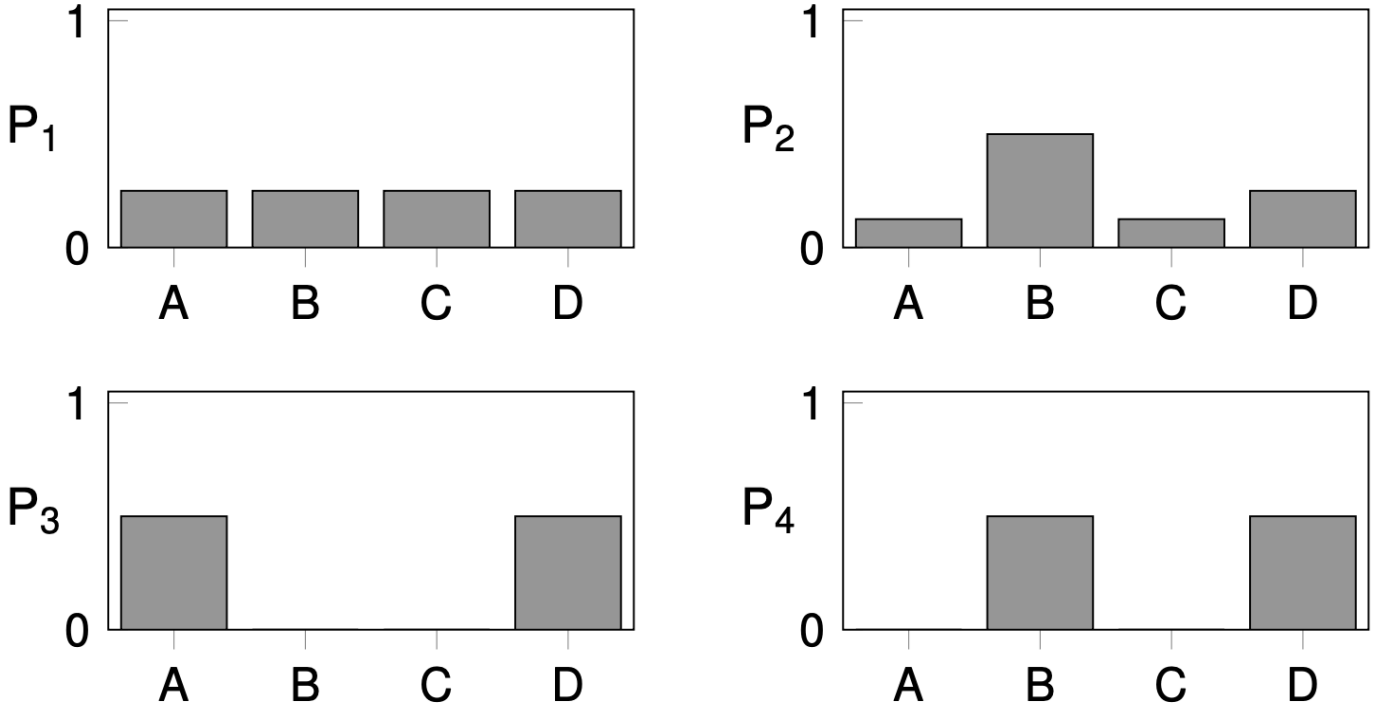
- Why D(p||q) isn't symmetric function ?

    - In the previous example of beta distributions, if **q(x)** is the "real" model, then a large fraction of samples drawn from **q(x)** will be able to strongly indicate that **p(x)** isn't a good model for generating those points. However, when you have **p(x)** as the real model, then the fraction of samples drawn from **p(x)** can't indicate as strongly that **q(x)** is a poor model. This is why the KL divergence isn't commutative.

Statistical Aspects of KL divergence

- Calculate these KL divergences:

$D_{\mathrm{KL}}(P_3 \| P_1)$ $D_{\mathrm{KL}}(P_2 \| P_4)$ $D_{\mathrm{KL}}(P_4 \| P_2)$ $D_{\mathrm{KL}}(P_3 \| P_4)$



Mutual Information

- The mutual information between two random variables, X and Y, is the divergence of the product of their marginal distributions from their actual joint distribution:

$$I[X;Y] \equiv D(\mathcal{L}(X,Y) \| \mathcal{L}(X) \times \mathcal{L}(Y))$$

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right)$$

- Mutual information captures dependency between random variables and is more generalized than vanilla correlation coefficient, which captures only the linear relationship

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} = \underbrace{H(Y)}_{\substack{\text{Uncertainty} \\ \text{about } Y}} - \underbrace{H(Y|X)}_{\substack{\text{Uncertainty} \\ \text{about } Y \\ \text{given } X}}$$

Mutual Information

- Commonly used in feature selection Mutual independence of zero guarantees that random variables are independent but zero correlation does not.
- Example applications in Bayesian networks, infoGAN …etc
- MI is symmetric:

$$I(X;Y) = I(Y;X)$$

- MI is non-negative:

$$I(X;Y) \geq 0 \quad , \quad I(X;Y) = 0 \text{ iff } X \perp\!\!\!\perp Y$$

- MI is a KL divergence:

$$I(X;Y) = D_{\mathrm{KL}}\left(p(x,y) \| p(x)p(y)\right)$$

Exercise

- Find the Entropy of a Gaussian distribution

Exercise

- Find the Entropy of a Gaussian distribution

$$H(X) = -\int_{-\infty}^{+\infty} \mathcal{N}(x|\mu, \boldsymbol{\Sigma}) \log \mathcal{N}(x|\mu, \boldsymbol{\Sigma}) dx$$

$$= \frac{1}{2}\mathbb{E}[\log |2\pi\boldsymbol{\Sigma}|] + \frac{1}{2}\mathbb{E}[(x-\mu)^\top \boldsymbol{\Sigma}^{-1}(x-\mu)]$$

$$= \frac{1}{2}\log |2\pi\boldsymbol{\Sigma}| + \frac{1}{2}\mathbb{E}[(x-\mu)^\top \boldsymbol{\Sigma}^{-1}(x-\mu)]$$

Exercise

- Find the Entropy of a Gaussian distribution

For the second term:

$$\mathbb{E}\left[\text{tr}\left((x-\mu)^\top \boldsymbol{\Sigma}^{-1}(x-\mu)\right)\right] = \text{tr}\left(\boldsymbol{\Sigma}^{-1}\mathbb{E}\left[(x-\mu)(x-\mu)^\top\right]\right)$$
$$= \text{tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}\right)$$
$$= D$$

Overall:

$$H(X) = \frac{1}{2}\log|2\pi e\boldsymbol{\Sigma}|$$

Information measures have analytical solutions for Gaussian distributions.

Statistical interpretation of information theory

- Assume we have data $\boldsymbol{x}_i \in \mathbb{R}^D$ generated from $p^*(\boldsymbol{x})$.
- Take family of models $p \in \mathcal{P} = \{p(\cdot|\boldsymbol{\theta}) : \boldsymbol{\theta} \in \mathbb{R}^M\}$.
- Assume there exists a $\boldsymbol{\theta}^*$ such that $p(\boldsymbol{x}|\boldsymbol{\theta}^*) = p^*(\boldsymbol{x})$.
- Consider maximum-likelihood estimator $\boldsymbol{\theta}_{\mathrm{ML}} = \operatorname*{argmax}_{\theta} \mathbb{E}[p(\boldsymbol{x}|\boldsymbol{\theta})]$. Then:

$$\mathbb{E}[\log p(\boldsymbol{x}|\boldsymbol{\theta}_{\mathrm{ML}})] = \mathbb{E}[\log p(\boldsymbol{x}|\boldsymbol{\theta}^*)] = -H[p^*(\boldsymbol{x})]$$

Entropy is the negative log-likelihood of the best model!

Statistical interpretation of information theory

Maximising likelihood is equivalent to minimising KL!

$$\boldsymbol{\theta}_{\mathrm{ML}} = \underset{\theta}{\operatorname{argmax}}\, \mathbb{E}[p(\boldsymbol{x}|\boldsymbol{\theta})]$$

$$\boldsymbol{\theta}_{\mathrm{ML}} = \underset{\theta}{\operatorname{argmin}}\, D_{\mathrm{KL}}(p^*(\boldsymbol{x}) \| p(\boldsymbol{x}|\boldsymbol{\theta}))$$

RECAP

- ✓ Entropy measures *uncertainty* or *randomness*.
- ✓ KL divergence measures *differences* between distributions.
- ✓ MI measures *correlation* between variables.

References

- A mathematical theory of communication", Claude E. Shannon, 1948
- Aurélien Géron, Introduction to Entropy, Cross-Entropy and KL-Divergence
- Goodfellow,Ian,YoshuaBengio,andAaronCourville.*Deeplearning*.MITpress,2016.
- Marko Cotra, Making sense of the Kullback–Leibler (KL) Divergence.
- Pedro A.M. Mediano, INFORMATION THEORY, INFERENCE, AND BRAIN NETWORKS
- Francois Fleuret, AMMI – Introduction to Deep Learning 5.1. Cross-entropy loss
- Shannon Entropy and Kullback-Leibler Divergence CMU statistics
- Abhishek Parbhakar, Must know Information Theory concepts in Deep Learning (AI)