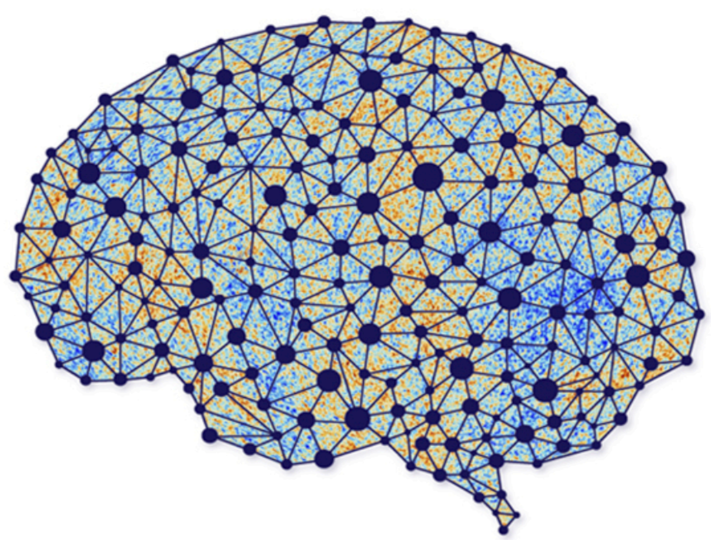
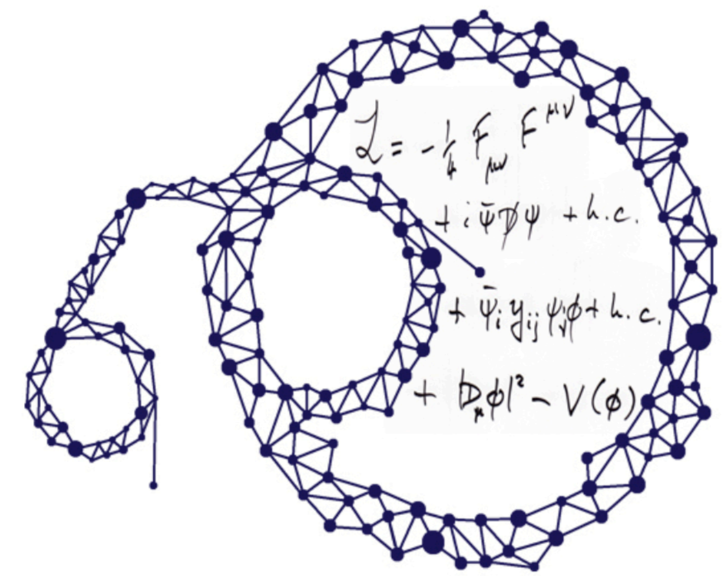
Physics 361 - Machine Learning in Physics

Lecture 4 – Background and Basics of Machine Learning

Feb. 1st 2024



AI
∩
Universe



Moritz Münchmeyer

Unit 1: Background

1.3 Information theory background

What is information theory?

- Quantify how much information is in a given signal.

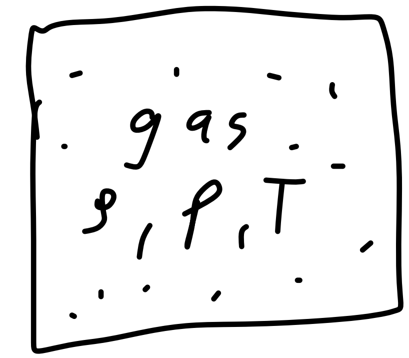
  → Optimal compression

- The information in a signal depends on the probability of the signal.
  - A certain signal/event contains no information.
  - A highly unlikely event contains a lot of information.

- In machine learning, information theory is used to characterize PDFs and their similarity.

Entropy

- In statistical physics (thermodynamics) the entropy is given by



$$S = (k_B) \log \Omega$$

here we set it to 1

Ω: number of microstates (equally likely)

- We can re-write this as

$$S = -\log p \quad \text{where} \quad p = \frac{1}{\Omega}$$

is the probability of each m. state

- The general definition of entropy

Shannon entropy $\boxed{S = -\sum_i p_i \log p_i}$

- log here is base e $\rightarrow$ entropy is in „nats"

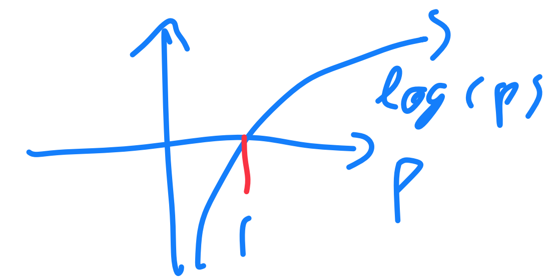
(base 2 $\rightarrow$ „ in bits)

Properties of the entropy

- discrete PDF $\quad S = -\sum_i p_i \underbrace{\log(p_i)}$

$-\log(p_i)$ is the „self-information" of event i

$$= -\mathbb{E}_{x \sim p(x)}\left[\log(p(x))\right]$$



A unlikely event has a large self info.

self information $\quad I(x) = -\log(p(x))$

A certain event has self inform. 0

maximum possible entropy is $S = \log(N)$

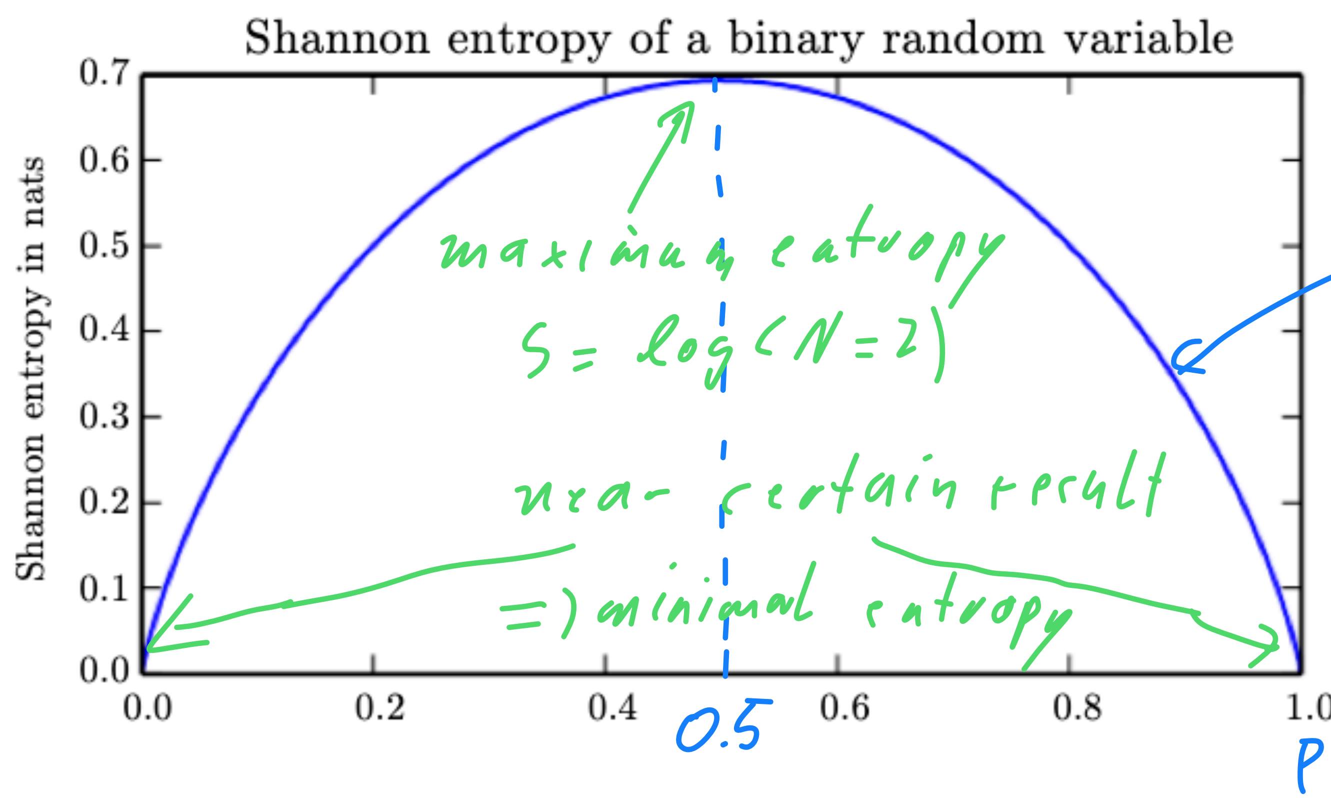
- continuous PDF

$$S = -\int dx\, p(x) \log(p(x))$$

$$= -\mathbb{E}_{x \sim p(x)}\left[\log(p(x))\right]$$

More uniform distributions have a higher entropy.
(spread out)

Example: Shannon entropy of a binary variable



entropy of the PDF as a fct of p.

deeplearningbook.org

Example: binary random variable $X : \begin{cases} 1 & : p \\ 0 & : 1-p \end{cases}$

calculate the entropy

$$S = - \sum_i p_i \log(p_i)$$

$$= - (1-p) \log(1-p) - p \log(p)$$

Kullback - Leibler divergence

- The relative entropy = KL divergence provides a measure of the similarity of two probab. distr. $P_{(x)}$ and $Q_{(x)}$:

difference of self information of sample x

$$D_{\mathrm{KL}}(P\|Q) = \mathbb{E}_{\mathrm{x}\sim P}\left[\log \frac{P(x)}{Q(x)}\right] = \mathbb{E}_{\mathrm{x}\sim P}\left[\log P(x) - \log Q(x)\right]$$

  - If P and Q are the same then $D_{KL} = 0$.

    e.g in generative ML

    $P(x)$: true (unknown) PDF of the training images

    $Q(x)$: PDF of generated images

- D_{KL} is easy to evaluate by sampling (if P and Q are known).
- $D_{KL} \geq 0$

- Interpretation: If our data is described by P and we have a theory/model Q that is meant to describe the data, at what rate will collection of data inform me that theory Q differs from P.

= KL divergence

- Minimizing the KL divergence is the same as maximizing the Likelihood of the data given the model Q. We define the cross-entropy as the part of the KL divergence that depends on the parameters of Q.

Cross entropy $H(P,Q) = -\mathbb{E}_{x\sim P(x)} \log[Q(x)]$

- The KL divergence is not a true distance metric. E.g it is not symmetric in P, Q.

Other alternatives : - Wasserstein distance
- F-divergence

- Another interesting concept: mutual information

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

vanishes if X, Y are independent.

Unit 2: Basics of Machine Learning

Sources: e.g. deeplearningbook.org

Overview

Most machine learning algorithms have the following elements:

- dataset
  - training data
  - test data
  - validation data
- cost function / loss function / training objective
- model / architecture
- optimization procedure

We first discuss these at the example of curve fitting (linear regression).

Unit 2: Machine Learning Basics

2.1 Machine Learning concepts using the example of Linear Regression

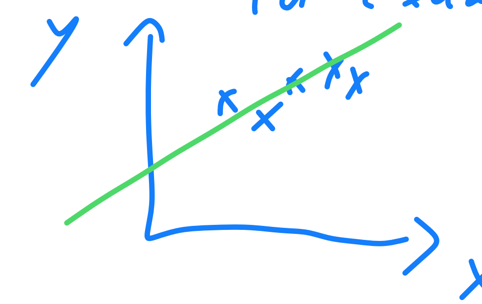
Linear regression

- The simplest machine learning algorithm.
- Predict 1 number from N features:

$$\hat{y} = \vec{w}^T \vec{x} + b$$

$$\vec{w}^T \vec{x} = \vec{w} \cdot \vec{x}$$

1d example



This can be re-written as

$$\hat{y} = \vec{w}^T \vec{x}$$

where we include b by adding an element 1 to x $\quad x = \begin{pmatrix} \vdots \\ 1 \end{pmatrix}$

- notation: We define the design matrix as

$$\mathbf{X} = \begin{array}{cc} & \text{features} \\ \text{example} & \begin{pmatrix} | & | & | & | \end{pmatrix} \end{array}$$

$\mathbf{X}^{\text{train}}$: training data

$\vec{y}$: training labels

- training set $\{\mathbf{X}^{\text{train}}, \vec{y}^{\,\text{train}}\}$

test set $\{\mathbf{X}^{\text{test}}, \vec{y}^{\,\text{test}}\}$

- The typical cost function for such regression problems is the Mean Squared Error (MSE).

$$MSE = \frac{1}{m} \sum_{i}^{\text{training d.}} (\hat{y}_i - y_i)^2$$

prediction ($\hat{y}_i$) truth (y_i)

Outliers with large „residual" $\hat{y} - y_i$ matter more in the loss due to the square.

for linear regression: $\hat{y}_i = \vec{w}^T x_i$

- Notation: L_p norm of a vector is

$$||\vec{x}||_p = \left(\sum_i |x_i|^p \right)^{1/p}$$

Thus we can rewrite

$$MSE = \frac{1}{m} || \hat{\vec{y}} - \vec{y} ||_2^2$$

- Goal is to minimize the MSE.

train to minimize MSE_{train} wrt. $\vec{w}$.

→ hope MSE_{test} will also be small.

- For linear regression we can solve for $\vec{w}$ analytically:

(derivation not required for this course, only the result)

$$\nabla_{\boldsymbol{w}} \mathrm{MSE}_{\text{train}} = 0$$

$$\Rightarrow \nabla_{\boldsymbol{w}} \frac{1}{m} ||\hat{\boldsymbol{y}}^{(\text{train})} - \boldsymbol{y}^{(\text{train})}||_2^2 = 0$$

$$\Rightarrow \frac{1}{m} \nabla_{\boldsymbol{w}} ||\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - \boldsymbol{y}^{(\text{train})}||_2^2 = 0$$

$$\Rightarrow \nabla_{\boldsymbol{w}} \left(\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - \boldsymbol{y}^{(\text{train})}\right)^\top \left(\boldsymbol{X}^{(\text{train})}\boldsymbol{w} - \boldsymbol{y}^{(\text{train})}\right) = 0$$

$$\nabla_{\boldsymbol{w}} \left(\boldsymbol{w}^\top \boldsymbol{X}^{(\text{train})\top} \boldsymbol{X}^{(\text{train})} \boldsymbol{w} - 2\boldsymbol{w}^\top \boldsymbol{X}^{(\text{train})\top} \boldsymbol{y}^{(\text{train})} + \boldsymbol{y}^{(\text{train})\top} \boldsymbol{y}^{(\text{train})}\right) = 0$$

$$\Rightarrow 2\boldsymbol{X}^{(\text{train})\top} \boldsymbol{X}^{(\text{train})} \boldsymbol{w} - 2\boldsymbol{X}^{(\text{train})\top} \boldsymbol{y}^{(\text{train})} = 0$$

$$\Rightarrow \boxed{\boldsymbol{w} = \left(\boldsymbol{X}^{(\text{train})\top} \boldsymbol{X}^{(\text{train})}\right)^{-1} \boldsymbol{X}^{(\text{train})\top} \boldsymbol{y}^{(\text{train})}}$$

we need matrix multiplication and inversion.

Course logistics

- **Reading for this lecture:**
  - **For example:** Deeplearningbook.org chapter 3 and 5.
- **Problem set**: First problem set next week.