neural networks

statistics and data analysis (chapter 10)

Björn Malte Schäfer

Graduate School for Fundamental Physics
Fakultät für Physik und Astronomie, Universität Heidelberg

June 23, 2016

wrap up: Bayesian evidence

- probability $p(H|I)$ of a hypothesis H being true, given information I
- require
  - sum-rule: $p(H|I) + p(\bar{H}|I) = 1$
  - product-rule: $p(X, Y|I) = p(X|Y, I)p(Y|I)$
- Bayes' law:

$$p(X|Y, I) = p(Y|X, I) \times \frac{p(X|I)}{p(Y|I)} \tag{1}$$

  with the hypothesis X and the data Y. identify:
  - $p(X|Y, I)$ posterior
  - $p(Y|X, I)$ likelihood
  - $p(X|I)$ prior
  - $p(Y|I)$ evidence
- I is the model familty, X one specific model characterised by a parameter value

Bayesian evidence for comparing models

- look at Bayes' law

$$p(X|Y,I) = p(Y|X,I) \times \frac{p(X|I)}{p(Y|I)} \tag{2}$$

  with

  - Y data
  - X model: model choice in a model family
  - I model family, with Y as one element

- evidence:

$$p(Y|I) = \sum_X p(Y|I,X)p(X|I) \quad \text{or} \quad p(Y|I) = \int_\Omega \mathrm{d}\theta\, p(y|\theta,I)p(\theta,I) \tag{3}$$

  for a continuous parameter space

- need prior $p(\theta, I)$, from theory or previous experiments, look for consistency between experiment and prior
- compare two models by evidence ratio: complexity vs. ability to fit

Bayesian evidence: Jeffrey's scale

- Neyman-Pearson-lemma: likelihood ratio is the best way of comparing hypotheses
- in this context, the likelihood ratio is called Bayes-ratio
- let's write down the **Bayes-ratio** between two competing models I_1 and I_2

$$B = \frac{p(Y|I_1)}{p(Y|I_2)} \tag{4}$$

- the Bayes-ratio can be expressed by marginalisation over all possible parameter choices θ

$$p(Y|I_i) = \int \mathrm{d}\mu\, p(D|\theta)p(\theta|I_i) \tag{5}$$

- simple models are preferred, because they've more "likelihood within the prior"
- Jeffreys scale for B for making decisions concerning I_1 vs. I_2
- scale for the degree of confidence in a model is arbitrary

prosecutor's fallacy

- Roman law: everybody is innocent until proven guilty, degree of evidence must support the hypothesis of being guilty beyond a reasonable doubt: if the prosecutor fails in proving the guilt, one reverts to the null-hypothesis innocence
- prosecutor's fallacy: neglecting the prior if evidence is used in court
- E is some evidence, I is the state of an accused being innocent:
  - $p(E|I)$ probability of damning evidence if the person is innocent
  - $p(I|E)$ probability of being innocent despite the evidence
- $p(E|I) \neq p(I|E)$ for conditional probabilities, but rather

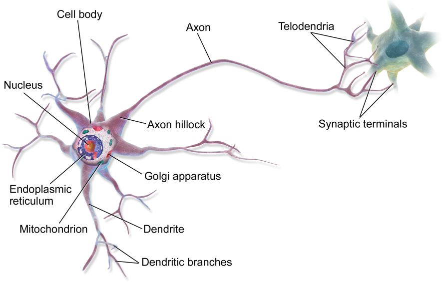
$$p(I|E) = p(E|I) \times \frac{p(I)}{p(E)} \tag{6}$$

  probability $p(E)$ of observing the evidence, and $p(I)$ being innocent
- $p(E|I)$ is tiny, but that does not mean that $p(I|E)$ is tiny as well!
- $p(E) = p(E|I)p(I) + p(E|\bar{I})(1 - p(I))$, with the wrong identification of an innocent person $p(E|I)$ and the identification of a guilty person $p(E|\bar{I})$

neural networks: idea

- up to now we
  - fitted physical models to data (regression) and
  - selected models based on simplicity and evidence
- this was motivated by a fundamental understanding of the laws of Nature
- but there might be applications where we may admit unphysical and complex models
  - effective description of data without underlying principles
  - difficult to understand classification tasks
- construct mathematical models with a high degree of complexity and flexibility
  - adjust all degree of freedom with known data
  - system might be able to abstract and perform well on similar data
  - without understanding the details

Ramon y Cajal: discovery of neurons



neuron (source: wikipedia)

- nerve cell is linked by synapses to other cells and form a network
- signal transmission is electrical (inside cells) and chemical (between cells)
- nice thought: evolution can't construct for a purpose, rather it uses methods which can adapt to model a certain behaviour

neurons in different animals

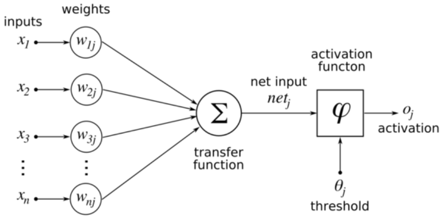
compare the number of neurons in different animals:

- sponge: 0
- jellyfish: 5×10^3
- fruit fly: 2.5×10^5
- frog: 1.6×10^7
- hamster: 9×10^7
- cat: 8×10^8
- human: 8×10^{10}
- elephant: 2.7×10^{11}

please check out

`https://en.wikipedia.org/wiki/List_of_animals_by_number_of_neurons`

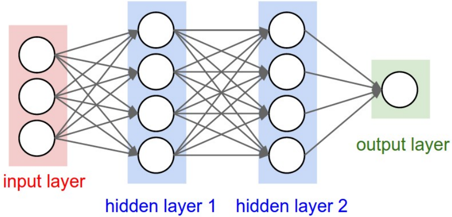
working principle of a neuron



artifical neuron (source: wikipedia)

- a neuron collects inputs x_i and computes a weighted sum $\sum_i w_i x_i$
- and compares the sum to a threshold θ: if $\sum_i w_i x_i > \theta$, it produces an output or an activation
- output is given by $\phi(\sum_i w_i x_i - \theta)$ with an activation function ϕ

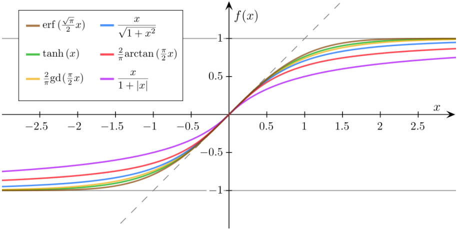
neural networks



network composed of artifical neurons

- neural networks are networks of many neurons
- neurons are arranged in layers
- representation theorem by Kolmogorov (different Kolmogorov): 2 layered networks can do every job, but there's no statement about how many nodes are needed
- if the information flux is from an input layer through hidden layers to an output layer, one refers to it as a feed-forward network

response functions



activation functions (source: wikipedia)

- many choices of the response function ϕ are possible
- usually one selects a monotonic, differentiable function, which asymptotes to constants
- the precise functional form usually does not matter a lot

classification by a neuron

- a neuron computes $\sum_i w_i x_i$ from the inputs and compares to the threshold θ
- think of w_i and x_i as a vectors: defines plane

$$w_i x_i - \theta = w_i x_i - w_i y_i \tag{7}$$

  with a normal vector w_i and a point y_i inside the plane
- $w_i x_i > \theta$ means that x_i lies above the plane, $w_i x_i < \theta$ below
- neuron defines a plane and quantifies if x_i lies above or below it
- response $\phi(w_i x_i - \theta)$ quantifies by how much
- application to a classification problem: find the best w_i and θ!

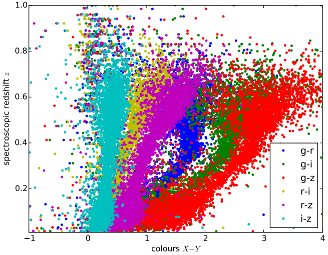
training as a fitting process

- adjust the every neuron's weights w_i and thresholds θ by requiring a minimised error on a training data set where the output is known
- technically, define the error as the difference between required and computed output,

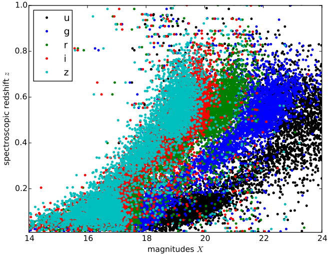
$$\Delta^2\left(\{w_i, \theta\}\right) = \sum_{\text{data}} \Delta_i^2 \tag{8}$$

  summed over all the training data
- Δ^2 is a function of all weights and thresholds
- backpropagation: weights are adjusted according to a gradient descent rule

example: estimating redshifts of galaxies



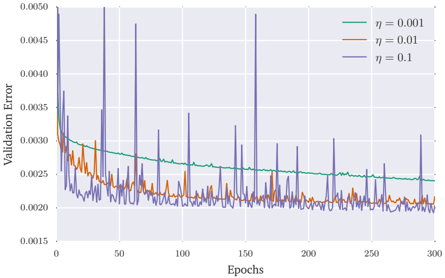
colours of galaxies in SDSS



magnitudes of galaxies in SDSS

- regression problem: estimation of the redshift based on colour or magnitude of a galaxy
- complicated relation, strong noise

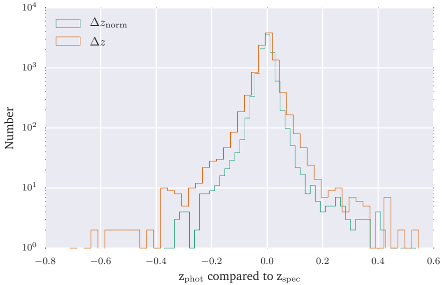
example: estimating redshifts of galaxies



error of the neural network on a verification sample (credit: L. Kiehl)

- use colours and magnitudes as input, estimate redshift
- compare estimated redshift to spectroscopic redshift

example: estimating redshifts of galaxies



distribution of the difference between estimated and true redshift (credit: L. Kiehl)

- distribution of the error shows a good accuracy of estimation

example: trying different networks

- finding the right solution involves a lot of skill and trial'n'error
- number of neurons and layers should be varied
- activation function, learning rate

please check out

`https://playground.tensorflow.org`

summary

- Bayesian evidence
  - test which model is prefered by data
  - complexity vs. simplicity, quantified by Bayes-evidence
  - very old idea, Occam's razor (scholastic era)
- neural networks
  - regression or classification with a very flexible but unphysical model
  - based on biological systems
  - difficult to understand what it actually does, but very powerful
  - new development: deep networks, with $\mathcal{O}(100)$ layers