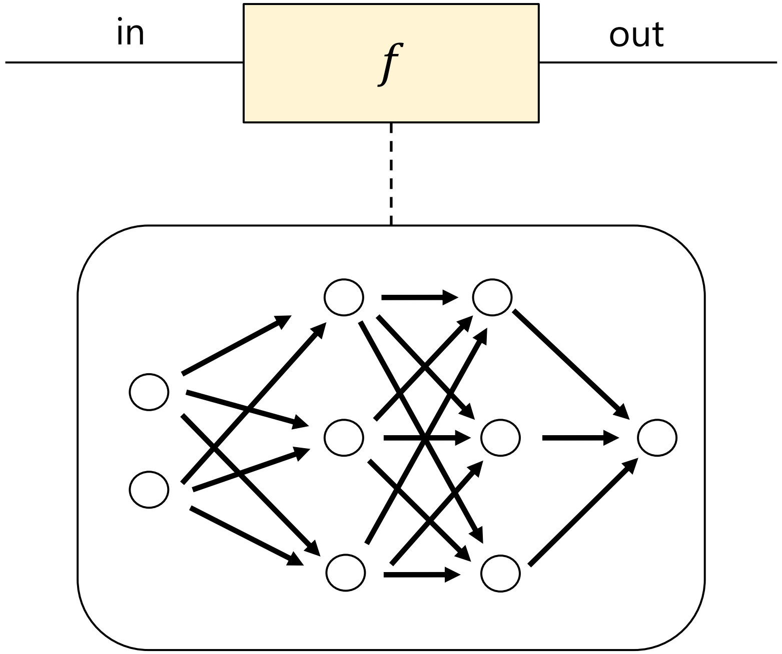
Some notes on Machine Learning

David Eklund

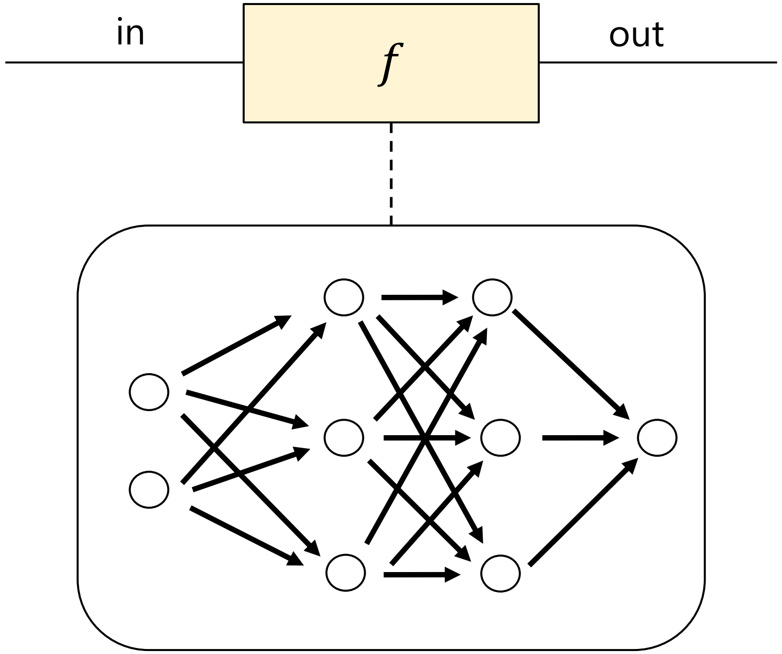
Machine learning and neural networks

- In machine learning, training data is used to learn a function (or a relationship).
- Input example: humidity, wind speed, pressure
- Output example: temperature
- For f we often use neural network. These are *function approximators* that can in principle mimic any function.
- Put differently, the network can describe many, many different functions and we need to find the right one (or a good one).



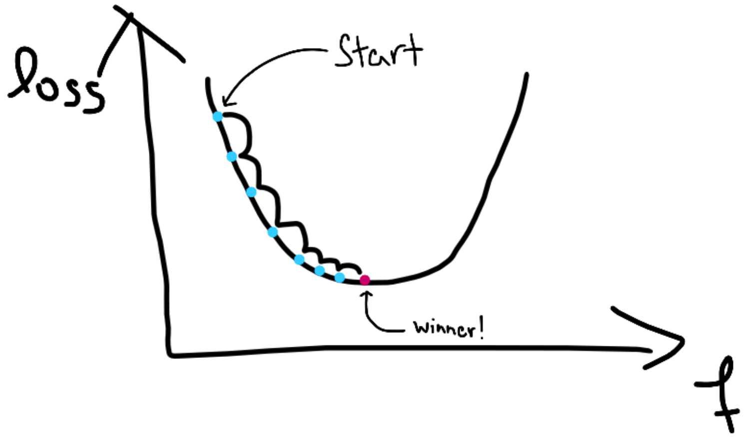
Machine learning and neural networks

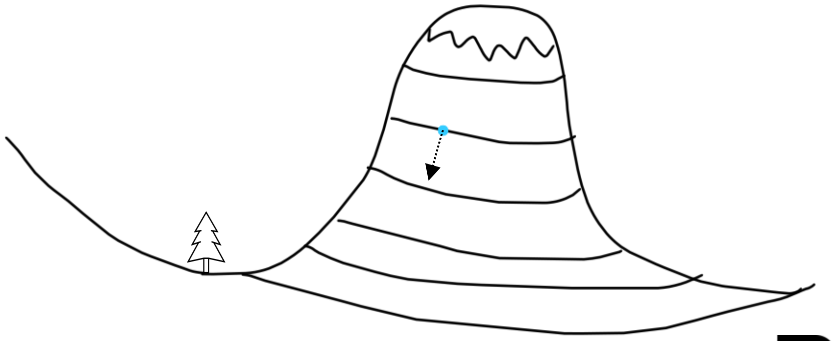
- The model depends on a number of adjustable parameters. The parameter values determine which function the model describes.
- It is these parameters that are adjusted during the model training.
- Neural networks are built as compositions of simpler parts in a network. Each layer is a linear (or affine) function together with a so-called activation function.
- Data commonly comes in (input, output) pairs.
- The network is supposed to learn from the training data examples and be able to generalize to new data.
- We often split in training and testing data so that the model is tested on data not used for training.



Machine learning and neural networks

- How good the network predictions are is measured with the loss function. Example: mean squared error.
- The task is to minimize the loss function.
- The learning procedure often goes something like this:
  - Start with some random guess of parameter values.
  - Repeat the following:
  1. Measure how good the network predictions are: evaluate the loss function.
  2. Modify the parameters incrementally to make a small improvement in the loss.
- Step 2 is gradient descent on the loss function: compute the derivate of the loss and take a step in the direction where it decreases the most.





Decision trees and random forests

- **Decision Tree Regression**: piece wise constant function used as model. The input space is partitioned into boxes where the model function has constant value.

- A greedy optimization algorithm is used to define the boxes (decide splitting variables and threshold values) and the model output on each box.

- **Random forests**: average the result of many decision trees trained on different parts of the data.

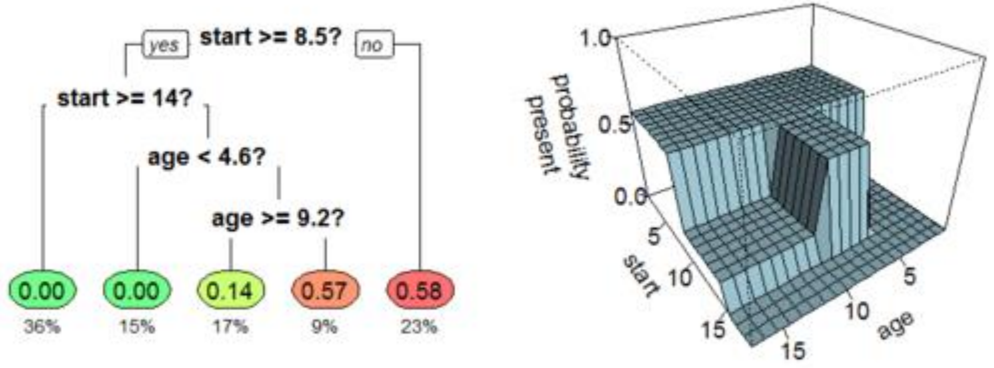


Image by Stephen Milborrow (cropped) distributed under licence https://creativecommons.org/licenses/by-sa/4.0/deed.en

Gaussian processes

- **Gaussian Process Regressors**: Stochastic processes (random functions) are used for regression.
- Incorporates Bayesian viewpoint.
- Includes confidence intervals (uncertainty of prediction) in the model.

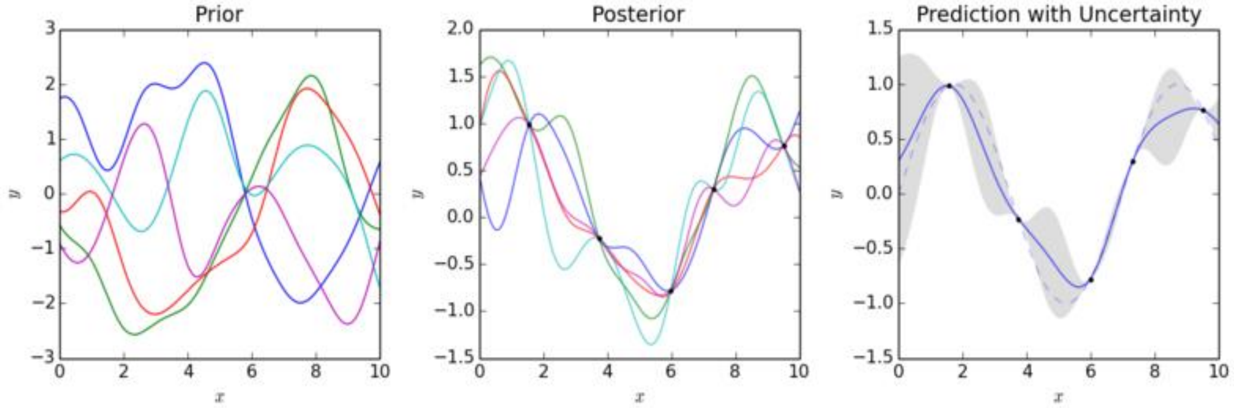


Image by Cdipaolo96 distributed under license https://creativecommons.org/licenses/by-sa/4.0/deed.en

Scientific Machine Learning

- Scientific machine learning mixes domain knowledge with machine learning algorithms.
- Model phenomena using both rules (or laws of nature) and data-driven techniques.
- Application 1: Our knowledge about a physical system is incomplete and needs completion through observational data.
- Application 2: Use machine learning as an efficient, robust, convenient, cheap or quick way of solving differential equations. Simultaneously fit to observational data.

UDEs, NODEs, NPDEs

Typical application of Universal Differential Equations (UDEs), Neural ODEs (NODEs) and Neural PDEs (NPDEs).

PINNs and other techniques

Typical application of Physics Informed Neural Networks (PINNs). Can also be used for Application 1.

Scientific Machine Learning

- Application 1: Can use a neural network to model the equation system.

- Example: Some conservation law or other property is known.

- Example: Some term of the equation system is known.

1. Encourage property to be fulfilled via the loss function.

2. Enforce known property via structure of the neural network.

3. Help neural network along the way with structure so that it needs to learn a less complicated function.

Neural network

$$\frac{\mathrm{d}y_1}{\mathrm{d}t} = KnownTerm_1 + NN(y_1, y_2, \dots)_1$$

$$\frac{\mathrm{d}y_2}{\mathrm{d}t} = KnownTerm_2 + NN(y_1, y_2, \dots)_2$$

$$\vdots$$

Can be done with UDEs/NODEs