

# Inverse problems and machine learning in medical physics

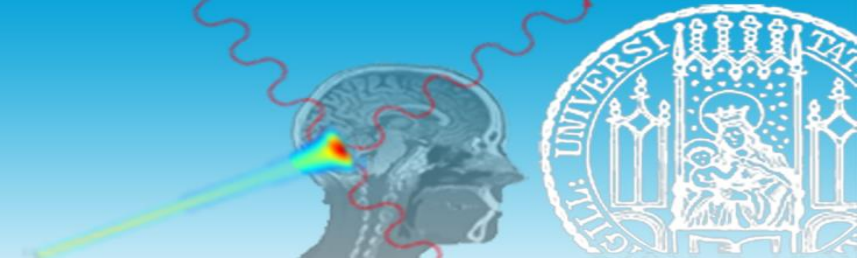
## Introduction to machine learning

Dr. Chiara Gianoli

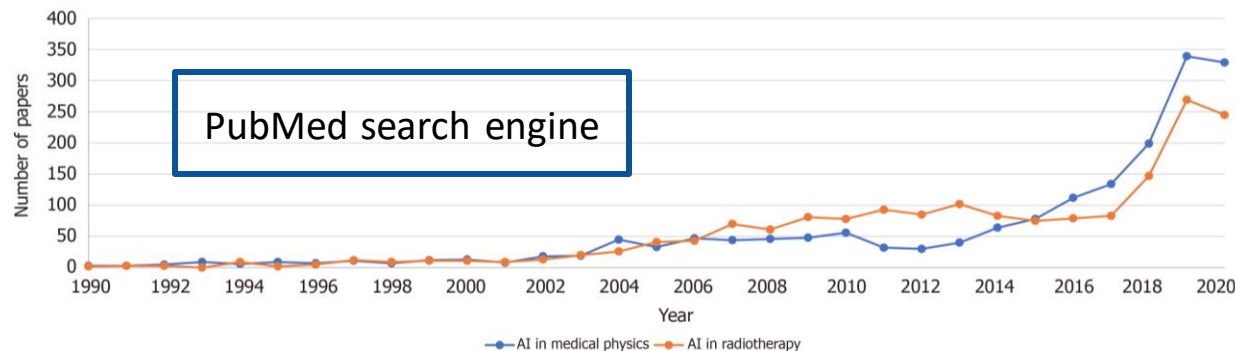
5/12/2023

[chiara.gianoli@physik.uni-muenchen.de](mailto:chiara.gianoli@physik.uni-muenchen.de)

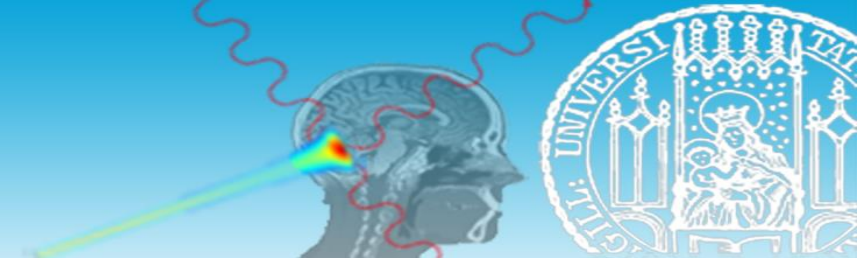
# Artificial intelligence and medical physics



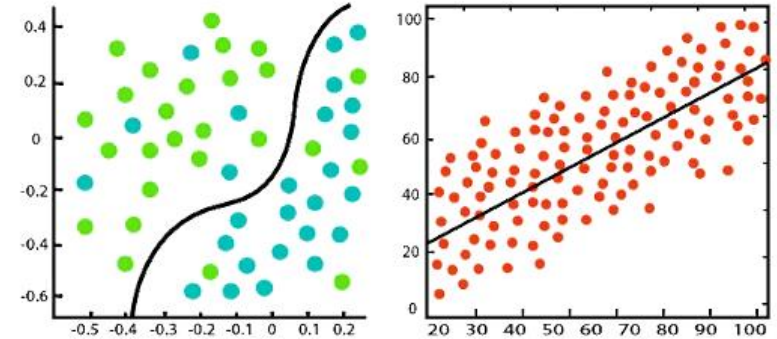
- The most remarkable developments in modern medicine have been related to the advances of **medical imaging**
- An unprecedented amount of **digital images** has been made available
  - The interest for **artificial intelligence** (AI) in medical imaging, with particular reference to **machine learning** (ML) and **deep learning** (DL), is growing enormously
  - In the near future, AI is expected to change profoundly healthcare and the application of physics to healthcare, referred to as **medical physics**



BioMed Central Medical Imaging



- The intent of AI (and ML/DL) is to **automate tasks** by turning data (i.e., **examples**) into models (i.e., **algorithms**)
- AI (and ML/DL) is applied to medical images for **problem solving** and **decision making** based on data to achieve a certain task
  - ML is typically based on **artificial neural network** (ANN)
    - Automatic extraction of hand-crafted features (i.e., “radiomic features”) for supervised or non-supervised **classification** (categorical output variable) or **regression** (continuous output variable)
  - ML based on **deep neural network** (DNN) is ML employing large-scale, multi-layer, hierarchical architectures, typically referred to as deep learning (DL)
    - Internally extracted features (i.e., inherent features or deep features) for supervised or non-supervised **prediction**

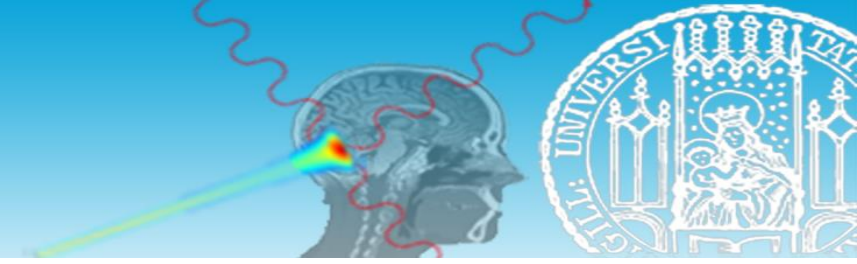


Classification

Regression

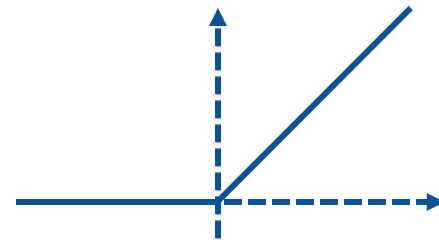
<https://www.javatpoint.com/regression-vs-classification-in-machine-learning>

# Fundamental concepts

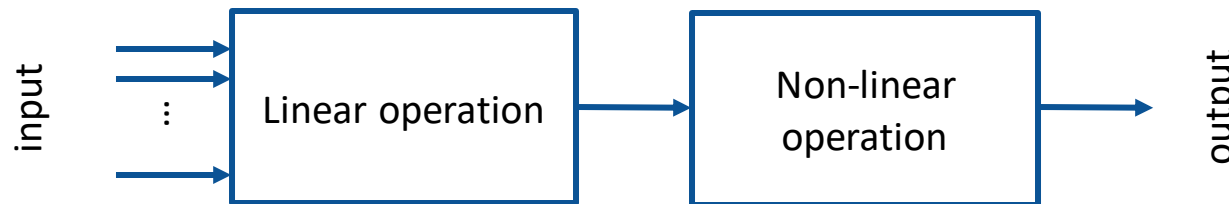


- The fundamental element of ANN and DNN is the **neuron** (i.e., the node of the network), designed in analogy to the **neuronal cell** (the biological neuron)
- The neuron consists of:
  - a group of input signals  $x_j$
  - a linear operation (i.e., weighted sum  $W_{ij}$  and bias  $b_i$ ) simulating **synaptic integration**
  - a non-linear operation, so called **activation function** (i.e., a rectified linear unit, ReLU) simulating the **action potential**

- 0 if the input value is negative
- the input itself if the input is positive

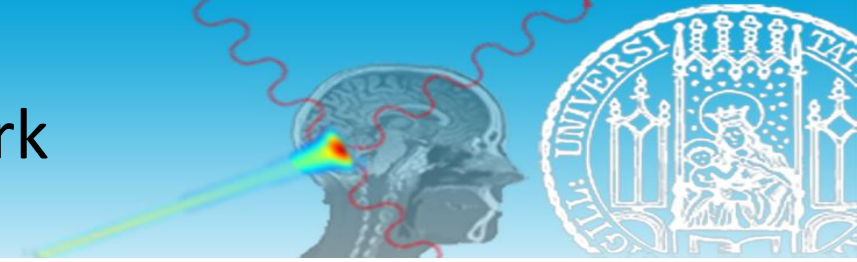


$$x_i = f\left(\sum_j W_{ij}x_j + b_i\right)$$

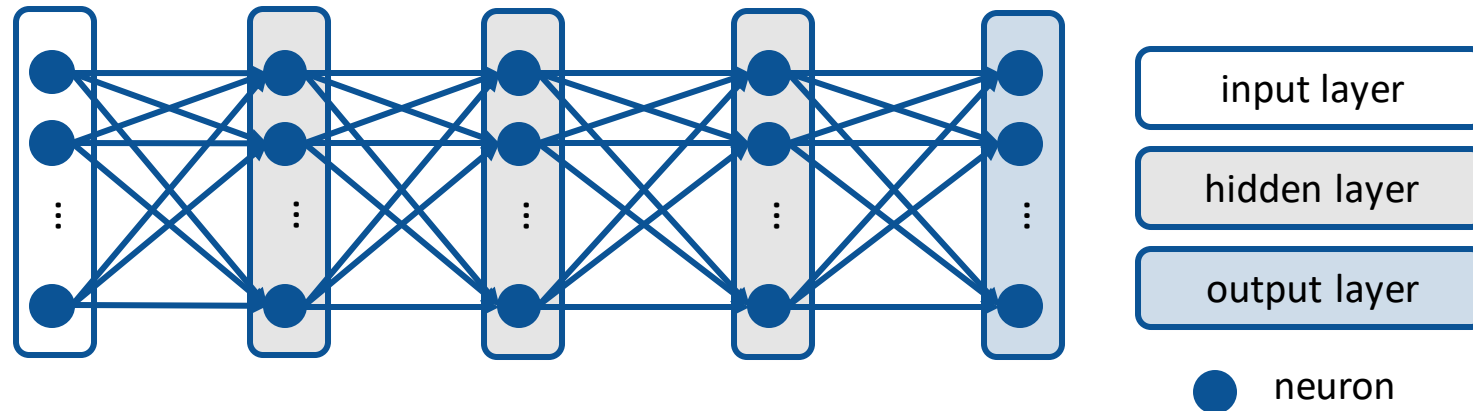


- The output of a neuron is fed to another neuron as one of the inputs

# Fully-connected neural network

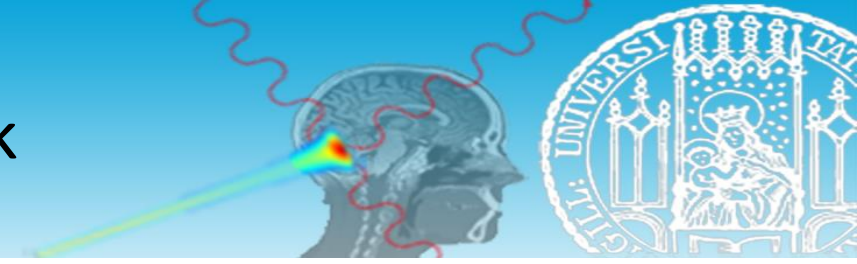


- The parameters of the network are defined by the learnable **weights** and **biases** of the **nodes** for each **layer**
  - A **layer** is a collection of nodes operating together at a specific **depth** within a neural network

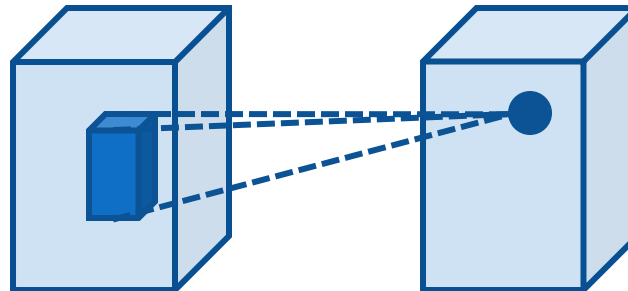


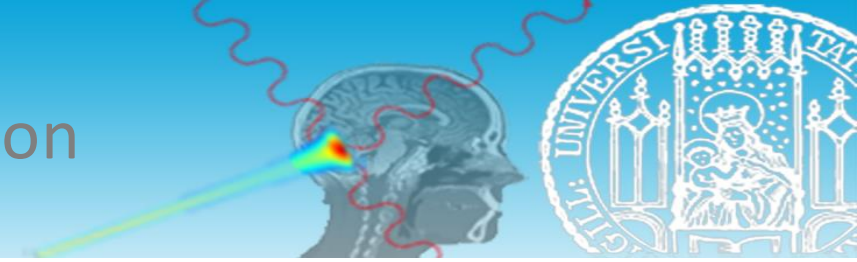
- The number of layers indicates the **depth** of the neural network
- The number of neurons in a layer indicates the **width** of the neural network

# Convolutional neural network



- Convolutional neural network (CNN) are designed to simulate the **visual nervous system**
  - The linear operation of the neuron becomes **image convolution** or actually **cross-correlation**
- The convolution is a weighted sum over a local region of the image so that the neuron of a layer is connected to this region of the preceding layer, thus reducing the number of parameters of the network
  - The parameters of the network are defined by the learnable kernels of the convolution
    - The **learnable kernels** represent the **receptors**
    - The **local region** represents the **receptive field** of the neuron





- **Convolution** calculates the result of a filtering operation between the image and a kernel (i.e., a filter)
- **Cross-correlation** calculates the similarity between the image and a kernel

- Cross correlation

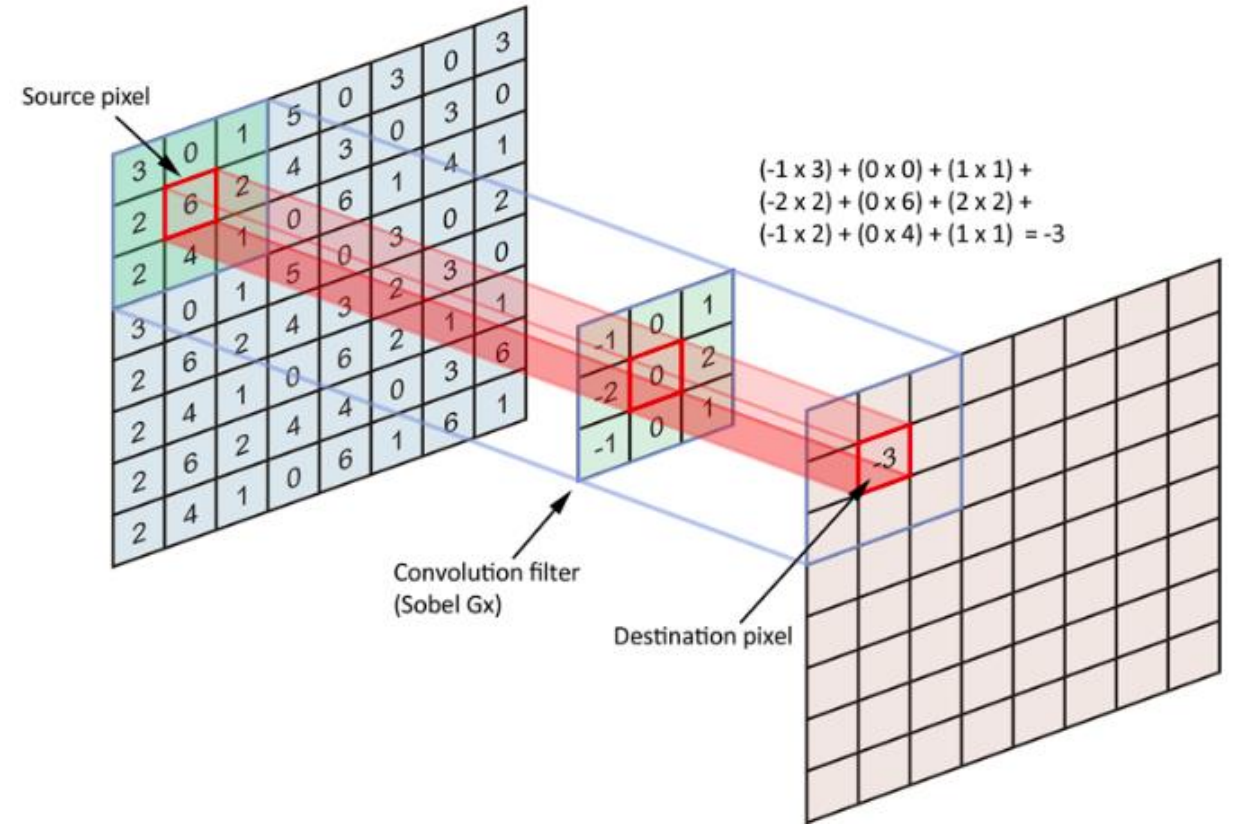
$$S[f] = w \otimes f$$

$$S[f](m, n) = \sum_{i=-k}^k \sum_{j=-k}^k w(i, j) f(m + i, n + j)$$

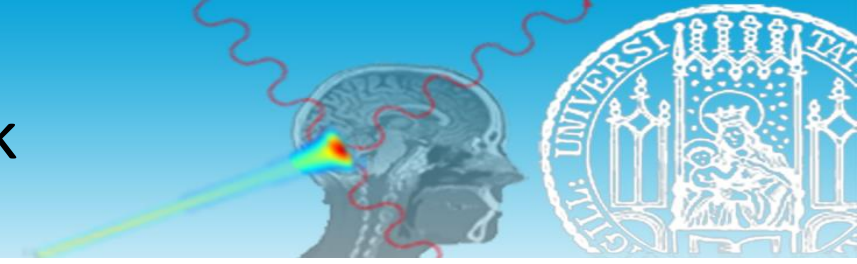
- Convolution

$$S[f] = w * f$$

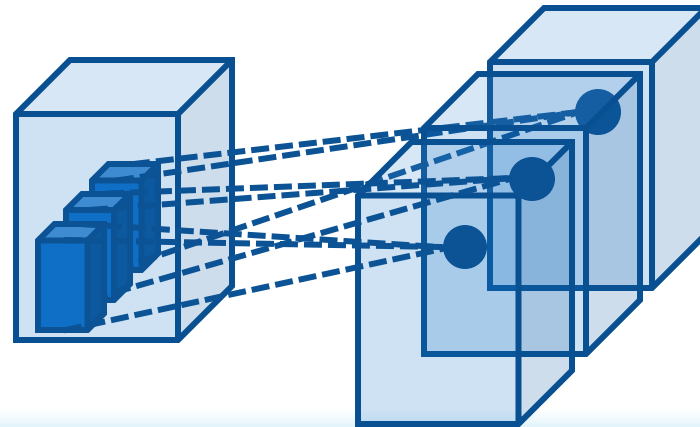
$$S[f](m, n) = \sum_{i=-k}^k \sum_{j=-k}^k w(i, j) f(m - i, n - j)$$



# Convolutional neural network

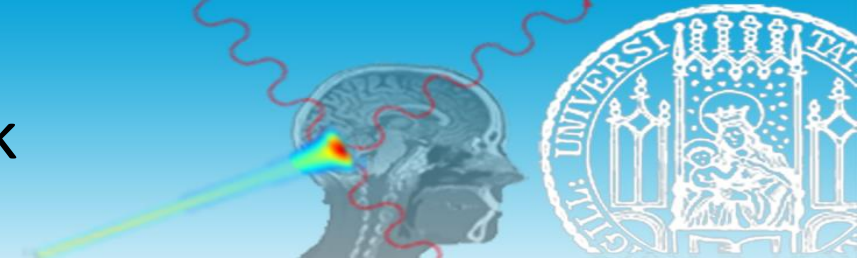


- The neurons of the CNN are organized according to the spatial dimensionality of the image (**height**, **width** for two-dimensional image) and to the number of kernels (**depth** or **channel dimension**)
- This mathematical object that generalizes vectors and matrices in a multi-dimensional array is referred to as **tensors**
  - The channels are intended to describe different aspects (**feature maps**) of the image
    - The input image itself can have different channels (i.e., RGB channels)
- Each **kernel** is applied onto the **input channel** of the previous layer to generate an **output channel**



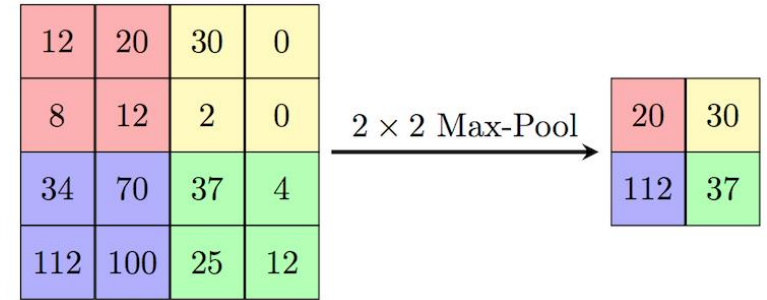


# Convolutional neural network

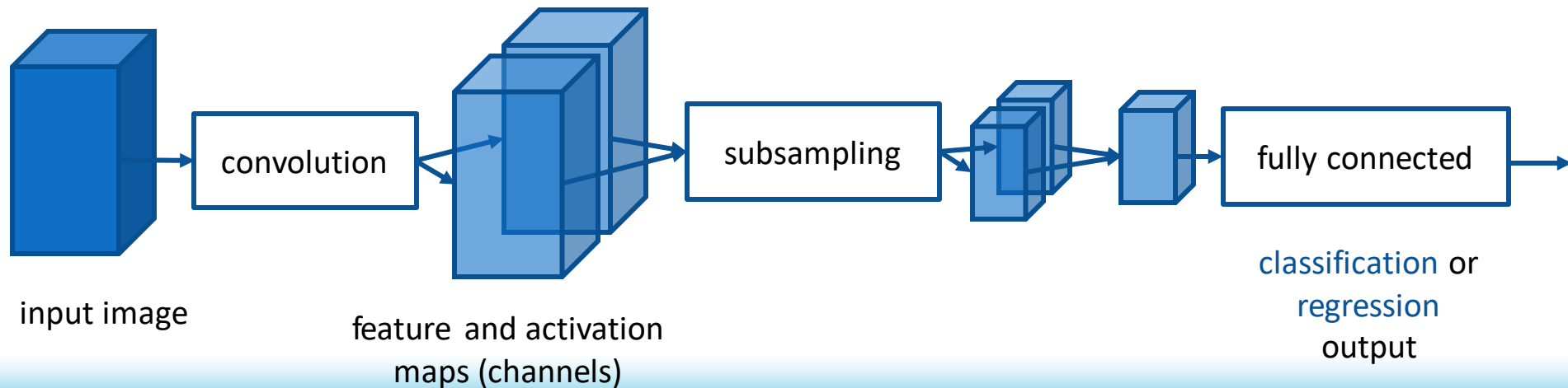


The CNN is typically composed by the stacking of three types of layers

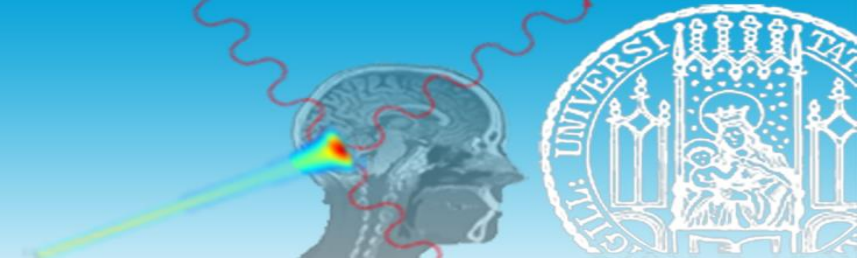
- **convolutional layers** → extracting features
- **pooling layers** → subsampling to remove redundancy and reduce the number of parameters (i.e., maximum pooling, average pooling...)
- **fully-connected layers** → adding non-linear and space variant combinations of these features (upon flattening)



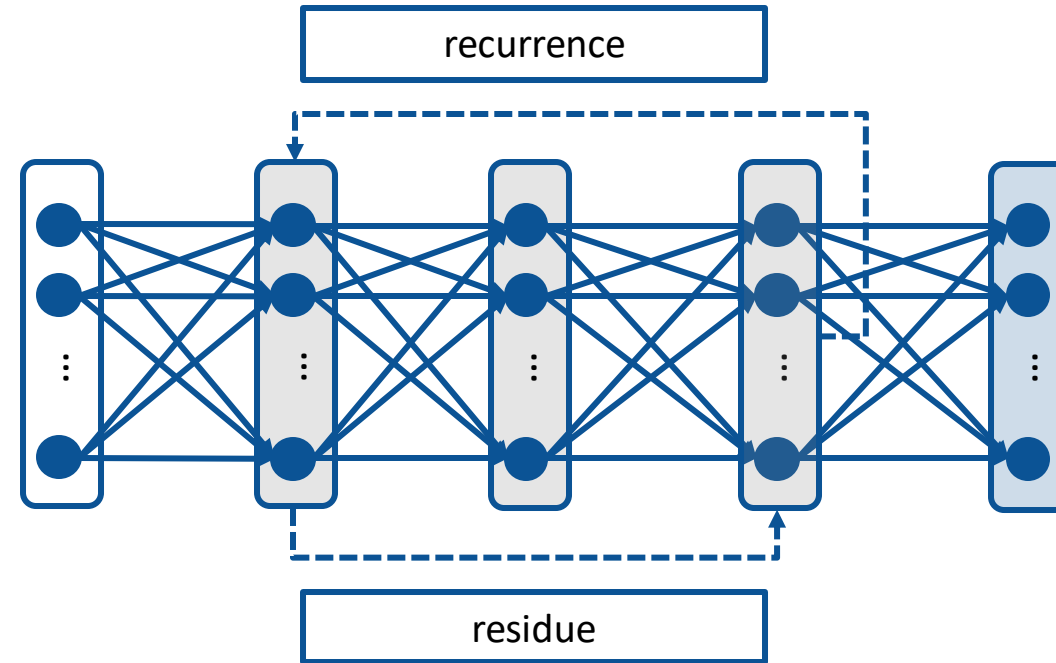
<https://paperswithcode.com/method/max-pooling>



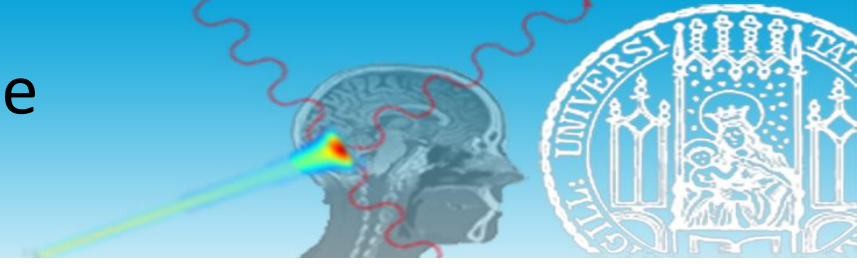
# Recurrent and residual neural networks



- In **recurrent neural networks**, outputs from the current layers are taken as inputs for the previous layer or the current layer itself (i.e., **feedback networks**)
  - Considering subsequent layers as a temporal sequence, recurrent neural networks process time-dependent inputs
  - A **fully recurrent neural network**, once unfolded through time, can be seen as a very deep feed-forward network in which all the layers share the same weights
- In **residual neural networks**, skip or residual connections are added to connect neurons in non-adjacent layers to preserve features as the network depth increases
  - Residual networks are an approximation of recurrent networks



# Mathematical formalism of the deep neural networks



- Input data are defined as  $x = [x^1, x^2, x^3, \dots]$
- Data at the hidden layer  $i$  are defined as  $h_i = [h_i^1, h_i^2, h_i^3, \dots]$
- Output data are defined as  $y = [y^1, y^2, y^3, \dots]$
- The linear and non-linear operations on the input of the hidden layer  $i$  are defined as  $f_i(h_i|\theta_i)$  being  $\theta_i$  the parameters describing these operations
- With a number of  $n$  layers, the DNN is written as:

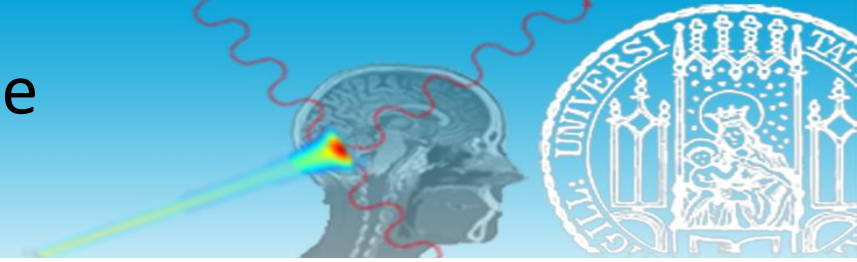
$$\begin{aligned}h_1 &= f_0(x|\theta_0) \\ h_i &= f_{i-1}(h_{i-1}|\theta_{i-1}) \\ y &= f_n(h_n|\theta_n)\end{aligned}$$

or as a [composite function](#):

$$y = f_n(f_{n-1}(f_{i-2}(\dots f_1(f_0(x|\theta_0)|\theta_1) \dots |\theta_{i-2})|\theta_{n-1})|\theta_n) = DNN(x|\theta)$$

that maps the input  $x$  to the output  $y$ , being  $\theta$  the entire set of parameters

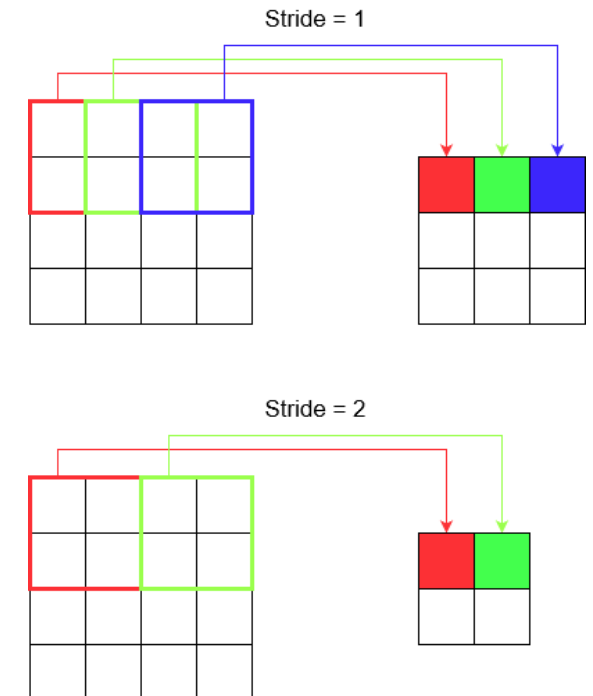
# Mathematical formalism of the deep neural networks



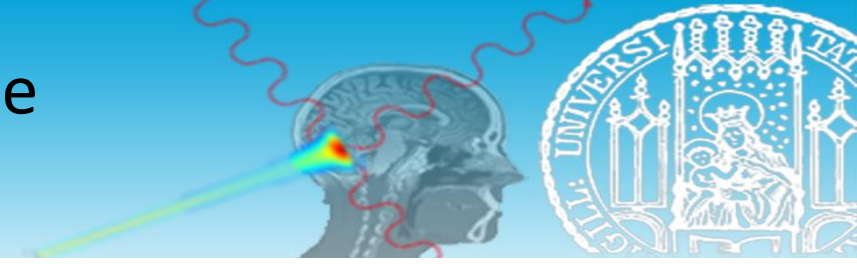
- Network training refers to the process of determining the parameters based on **training data**
  - The learnable kernels (weights and bias) are the **model parameters**
  - In CNN, kernel size (i.e., the size of the convolution), the stride (i.e., the density of the convolution) and the padding (i.e., border of the convolved image) are called **hyperparameters**
    - Pooling layer has no learnable parameter but only hyperparameters
- The training is mathematically formulated as solving an optimization problem where the goal is to find the **model parameters** that minimize a **loss function**

$$\hat{\theta} = \operatorname{argmin}_{\theta} L(\theta) = \operatorname{argmin}_{\theta} L(\text{DNN}(x|\theta))$$

- $L(\theta)$  is problem-specific



# Mathematical formalism of the deep neural networks



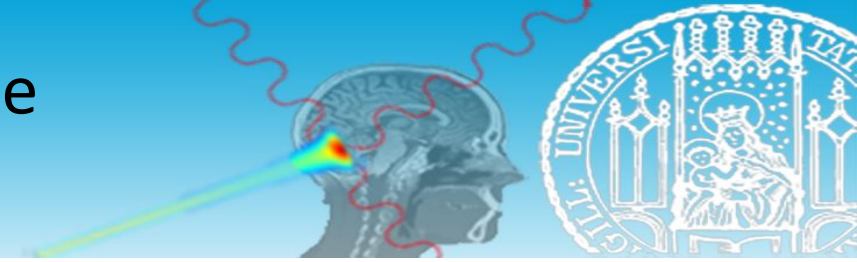
- The optimization is typically solved via **gradient descend algorithms** that update the parameters  $\theta$  as:

$$\theta^{k+1} = \theta^k - \lambda \nabla_{\theta} L(\theta)$$

where  $k$  is the iteration number,  $\nabla_{\theta} L$  is the gradient and  $\lambda$  is the learning rate

- When the evaluation of the gradient of the loss function is complicated (such as in complex DNN), the parameters  $\theta$  are updated according to **backpropagation**
- After each forward pass (i.e., from input to output of the network), the backpropagation performs the backward step (i.e., from output to input of the network) while computing the gradient of the loss function with respect to parameters at **each node** and for **each layer** (i.e., the **chain rule**)
  - The **backpropagation** adjusts each **weight** and **bias** in the network in proportion to how much they contribute to the loss function
  - The **weights** and **biases** are updated according to the negative direction of the gradient of the loss function, after each (forward pass and) backward pass

# Mathematical formalism of the deep neural networks



- The calculation of the gradient in one layer takes part of the calculation of the gradient in the previous layer
  - The derivative of a composite function is equal to the multiplication of the derivatives of those functions

$$\frac{\partial}{\partial x} (f(g(x))) = g'(x)f'(g)$$

$$\frac{\partial}{\partial x} (f(g(h(i(j(k(x))))))) = \frac{\partial k}{\partial x} \frac{\partial j}{\partial k} \frac{\partial i}{\partial j} \frac{\partial h}{\partial i} \frac{\partial g}{\partial h} \frac{\partial f}{\partial g}$$

chain rule

- The adjustment of a single **weight** can be intuitively explained by a simplified network made of 2 layers and 2 neurons

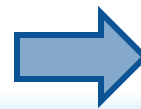
- Loss function  $L = (y - t)^2$  where  $t$  is the desired output,  $t = 0.5$



- For  $x = 1.5$ 

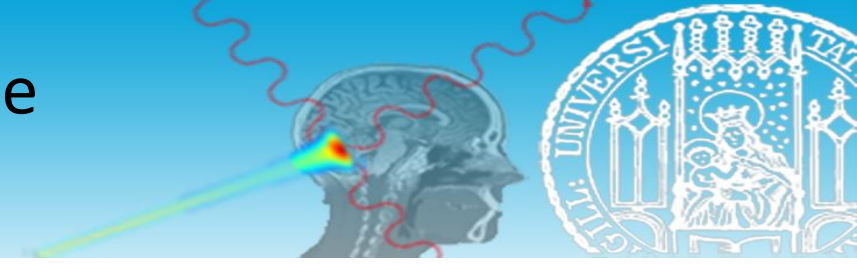
$$\frac{\partial L}{\partial y} = 2(y - t) = 2(1.5w) - 1$$

$$\frac{\partial y}{\partial w} = x = 1.5$$

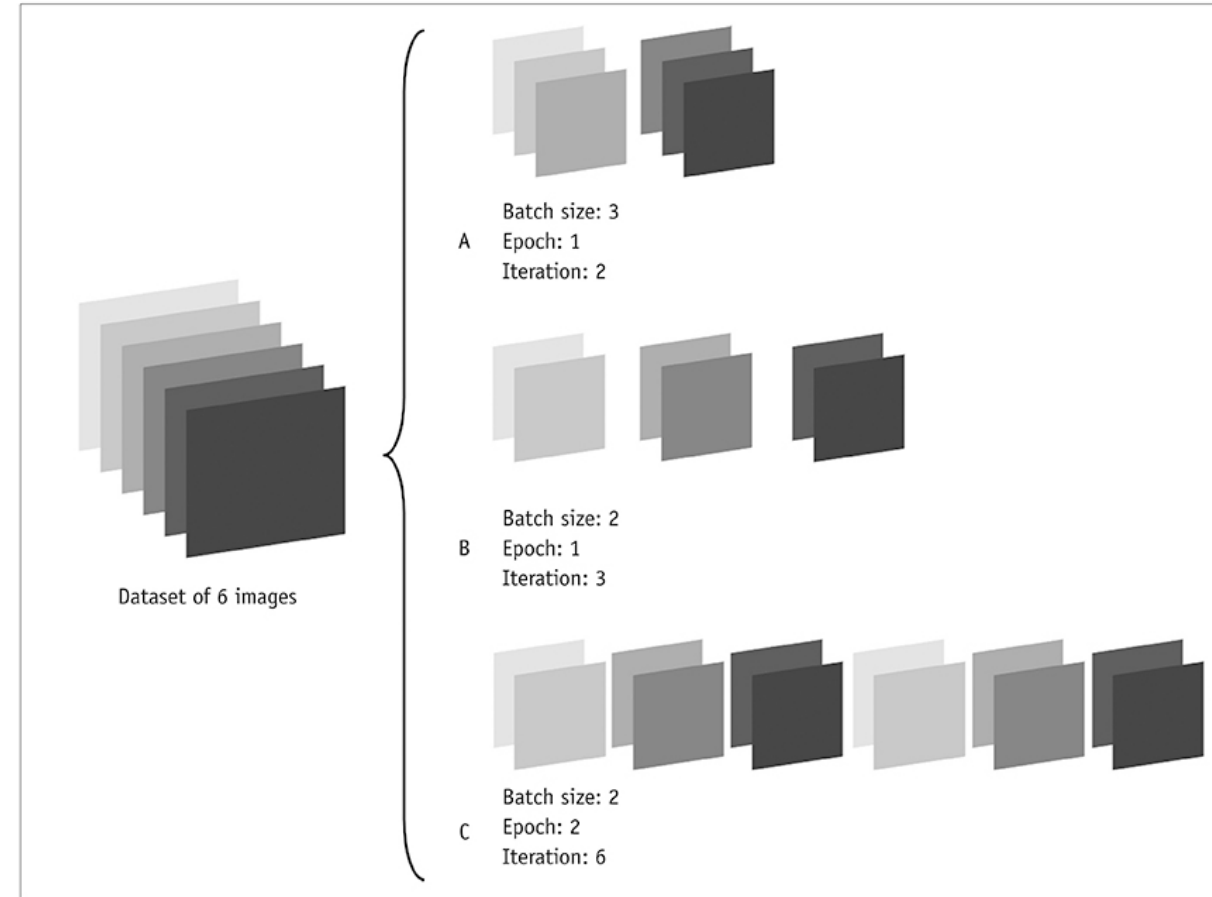


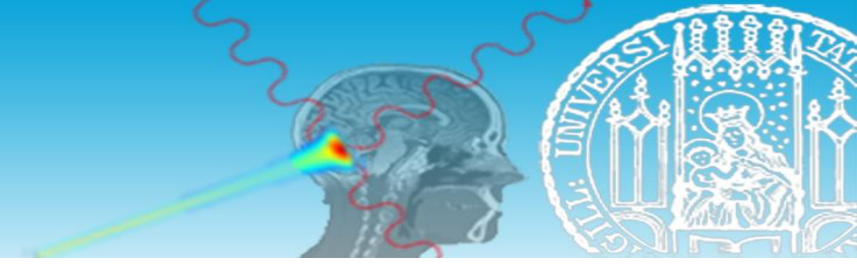
$$\frac{\partial L}{\partial w} = \frac{\partial y}{\partial w} \frac{\partial L}{\partial y} = 1.5(2(1.5w) - 1) = 4.5w - 1.5$$

# Mathematical formalism of the deep neural networks

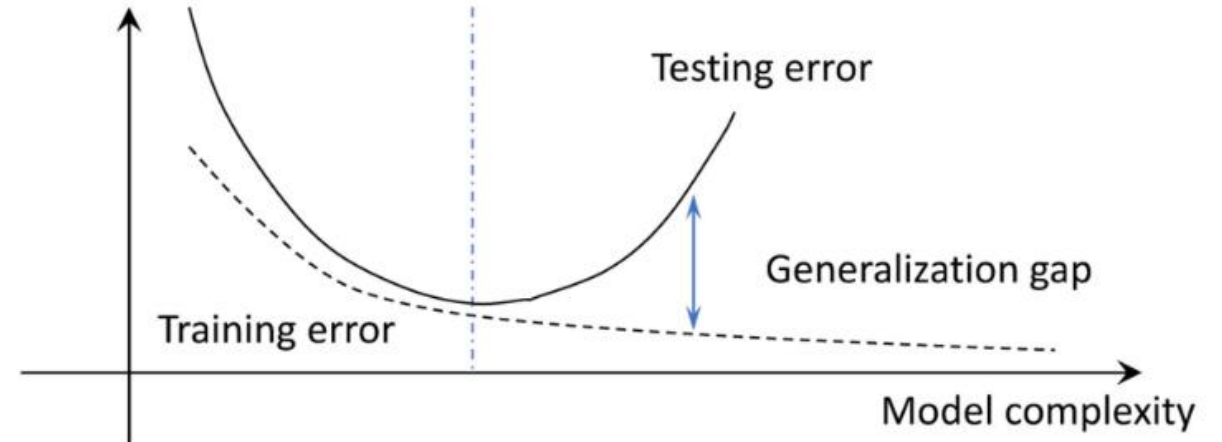


- The training data are shuffled and split into a number of small portions called **batches** to update the parameters
  - The backpropagation is applied to the mean of the loss functions in each **batch**
  - In one epoch, the training data of all batches are passed forward and backward through the network only once
- Learning rate, batch size and number of epochs are also **hyperparameters**



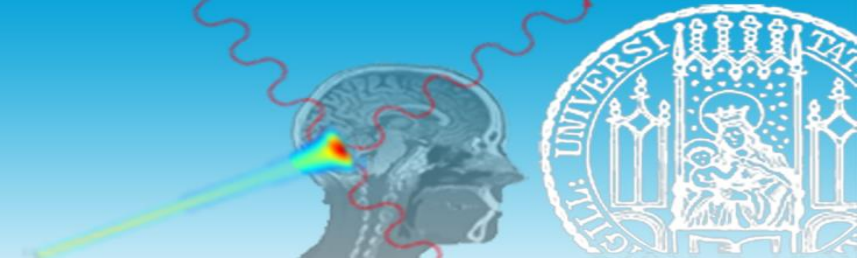


- After having trained the network (based on training data), the evaluation of the model is based on testing data
  - The data are split into training data and testing data (**hold-out**)
  - The data can be randomly split up into groups (i.e., folds) and iteratively, one group is adopted for testing and the others for training (**cross-validation**, i.e., k-fold cross-validation)
    - **Underfitting** - Training error and testing/validation error are high
    - **Overfitting** - Training error is low but testing/validation error is high
    - **Good fitting** - training error is low, slightly lower than the testing/validation error

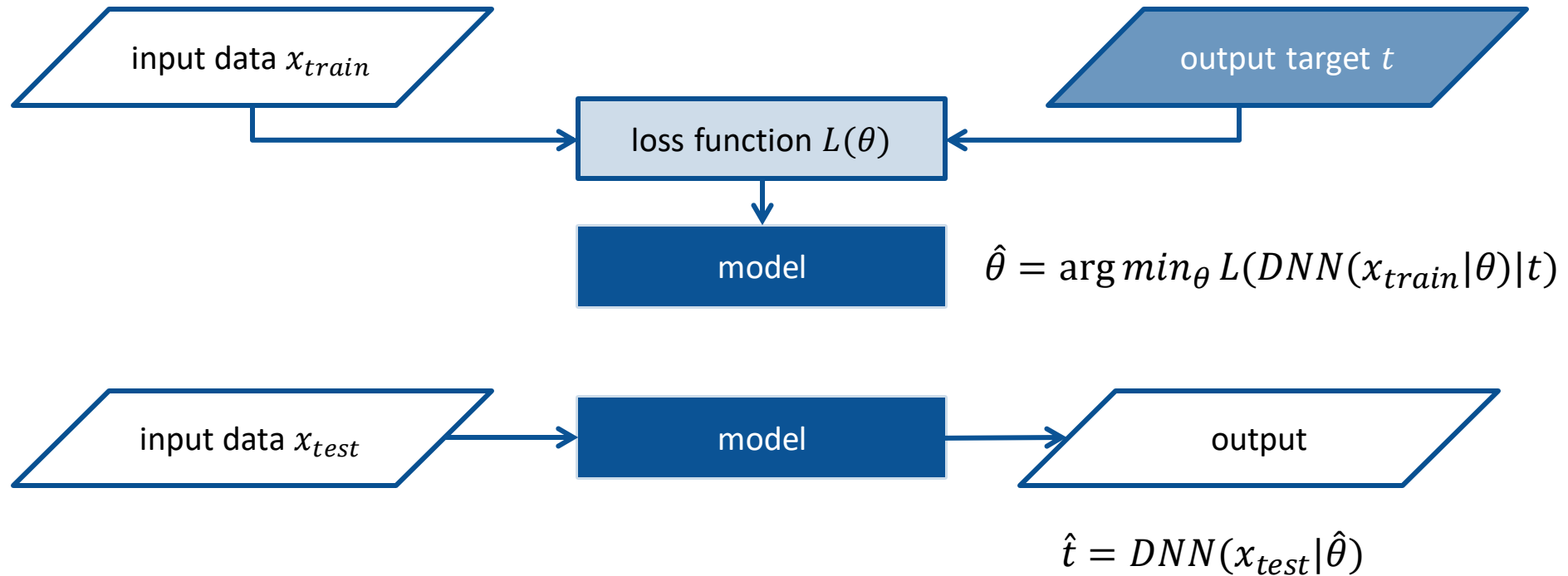




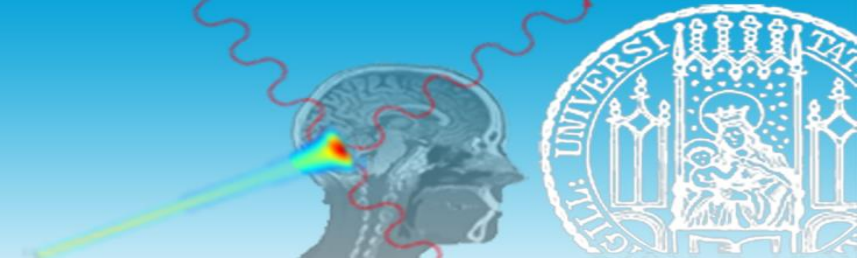
# Supervised learning



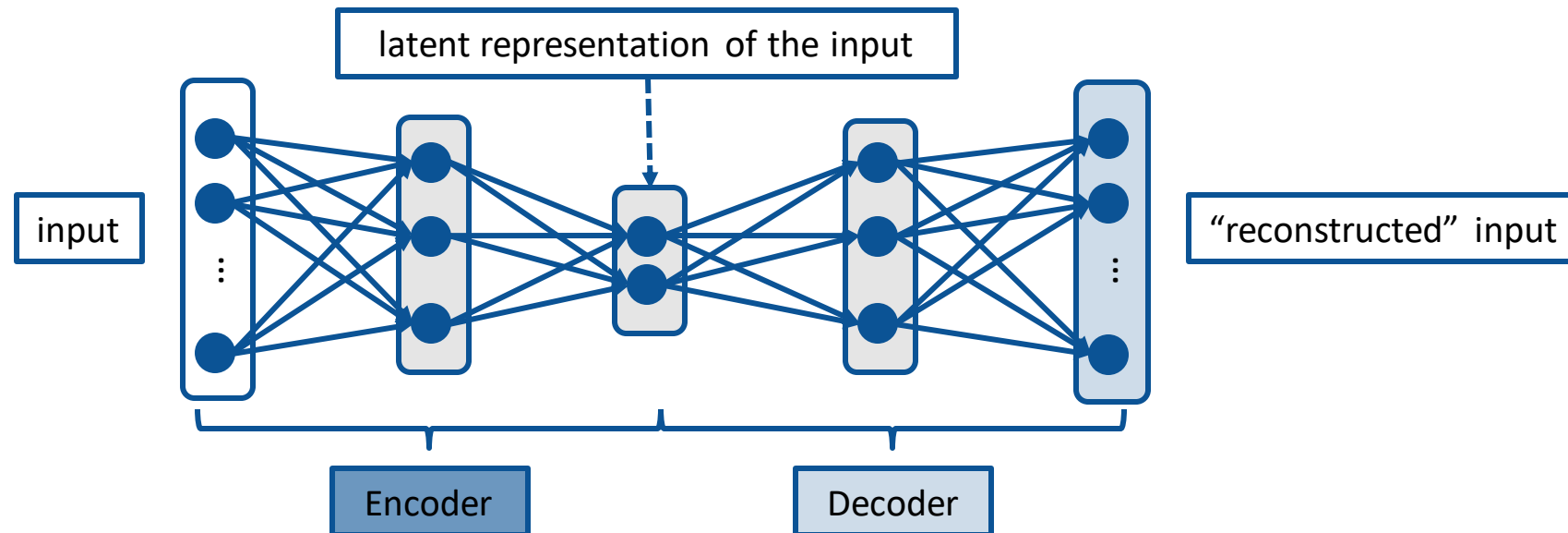
- In supervised learning the model is trained by explicitly enforcing the compliance of the model to paired input data and output target



- The availability of such a correspondence is a strict requirement

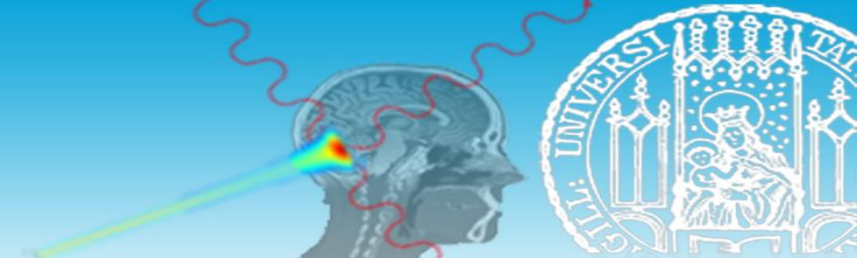


- In **unsupervised learning** the model is trained by purely relying on input data
  - The **deep auto-encoder** is an example of unsupervised machine learning, that can be based on CNN or fully connected DNN or a mixture of them
    - Two-way mappings between the **original data space** of the input and a **latent space** of a relatively lower dimension than the original data space

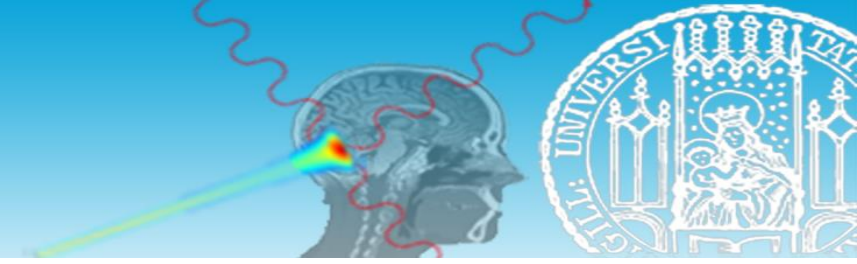


- The model is trained by simply enforcing the agreement between the input and the corresponding "reconstructed" input

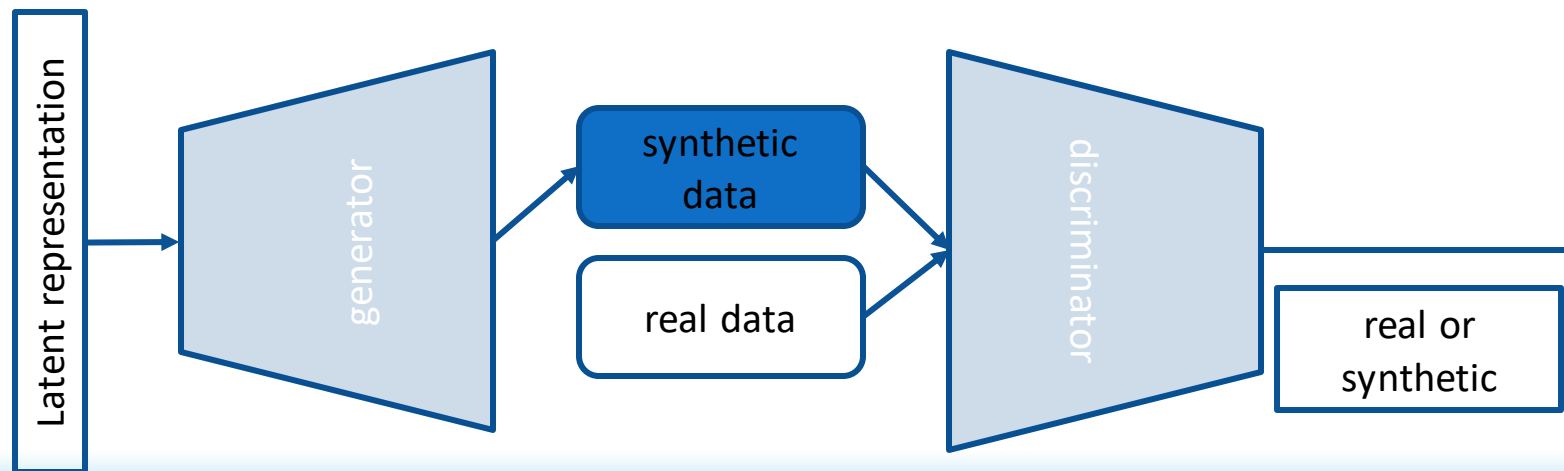
# Unsupervised learning



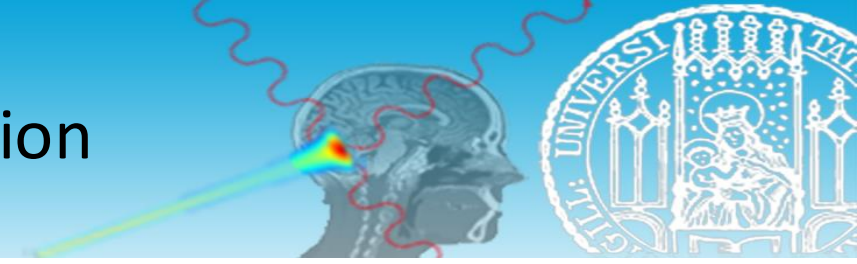
- The latent representation of the input preserves most of the information from the original data space (i.e., compressed information), such that the input can be exactly “reconstructed” based on the latent representation
  - The function that maps from the original data space to the latent space is called an **encoder**
    - The convolutional neural network maps the image to a collection of features in the feature space
  - A **decoder** stands for the mapping that “reconstructs” the data from the latent representation
    - The “deconvolutional” neural network, reversing the operations of the **convolutional neural network** by means of **deconvolution** (i.e., the inverse kernels) or “**transposed**”/reversed **convolution**, maps these features back to the image space
  - Since the **pooling** is non-invertible, the operation of **unpooling** is an approximation and the locations of the maximum/average are saved during pooling and used during unpooling



- The model is trained by relying on target information only partially available
- The **generative adversarial network** (GAN) is one of the most widely used
  - A generative network (**generator**, **encoder**) and a discriminative network (**discriminator**, **decoder**) are trained simultaneously to fight against each other
    - The **discriminator** is trained to distinguish real and synthetic samples
    - The **generator** is trained to produce examples that are realistic enough to fool the discriminator

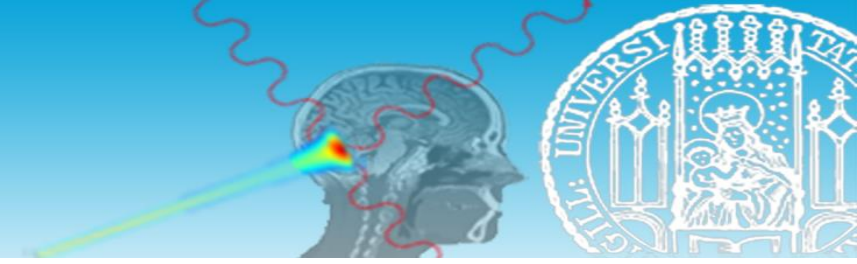


# Tomographic image reconstruction

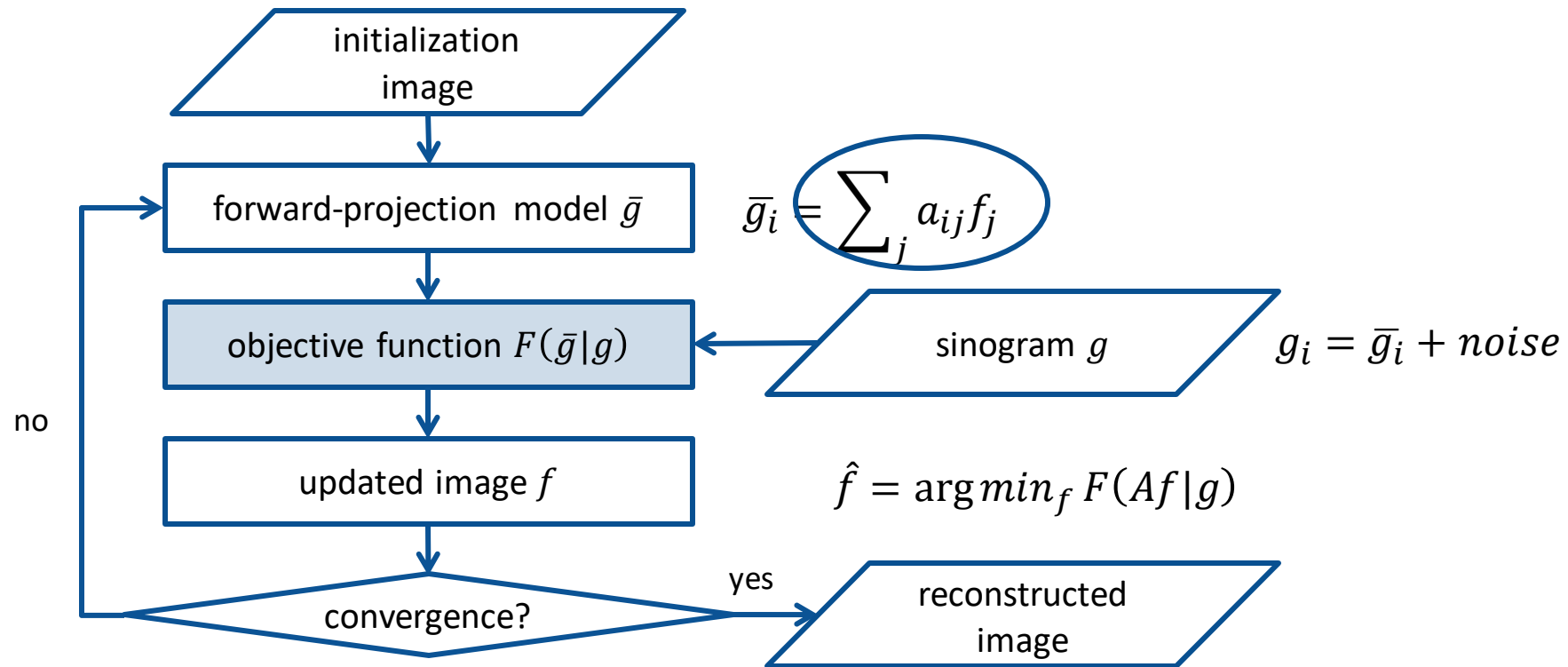


- Tomographic image reconstruction represents the building block of medical imaging
- Tomographic image reconstruction has been classified as **analytical reconstruction** or as **iterative reconstruction**
- Very recently, data-driven, deep-learning-based tomographic image reconstruction has been introduced (i.e., **deep tomographic reconstruction**)
  - Direct reconstruction methods
  - Unrolled iterative reconstruction methods
- The huge benefit of machine learning in reconstruction is the use of the **ground truth** (i.e., supervised learning), as obtained from high quality simulations or high quality measurements

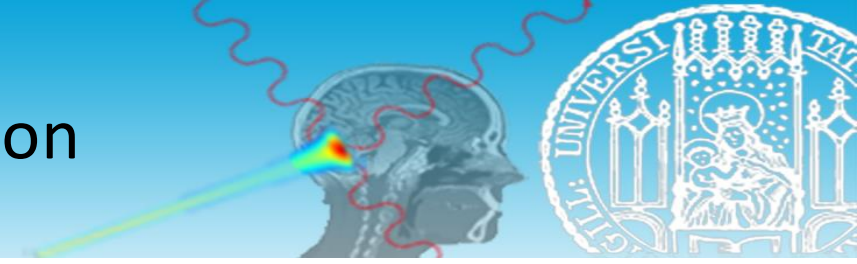
# Iterative reconstruction



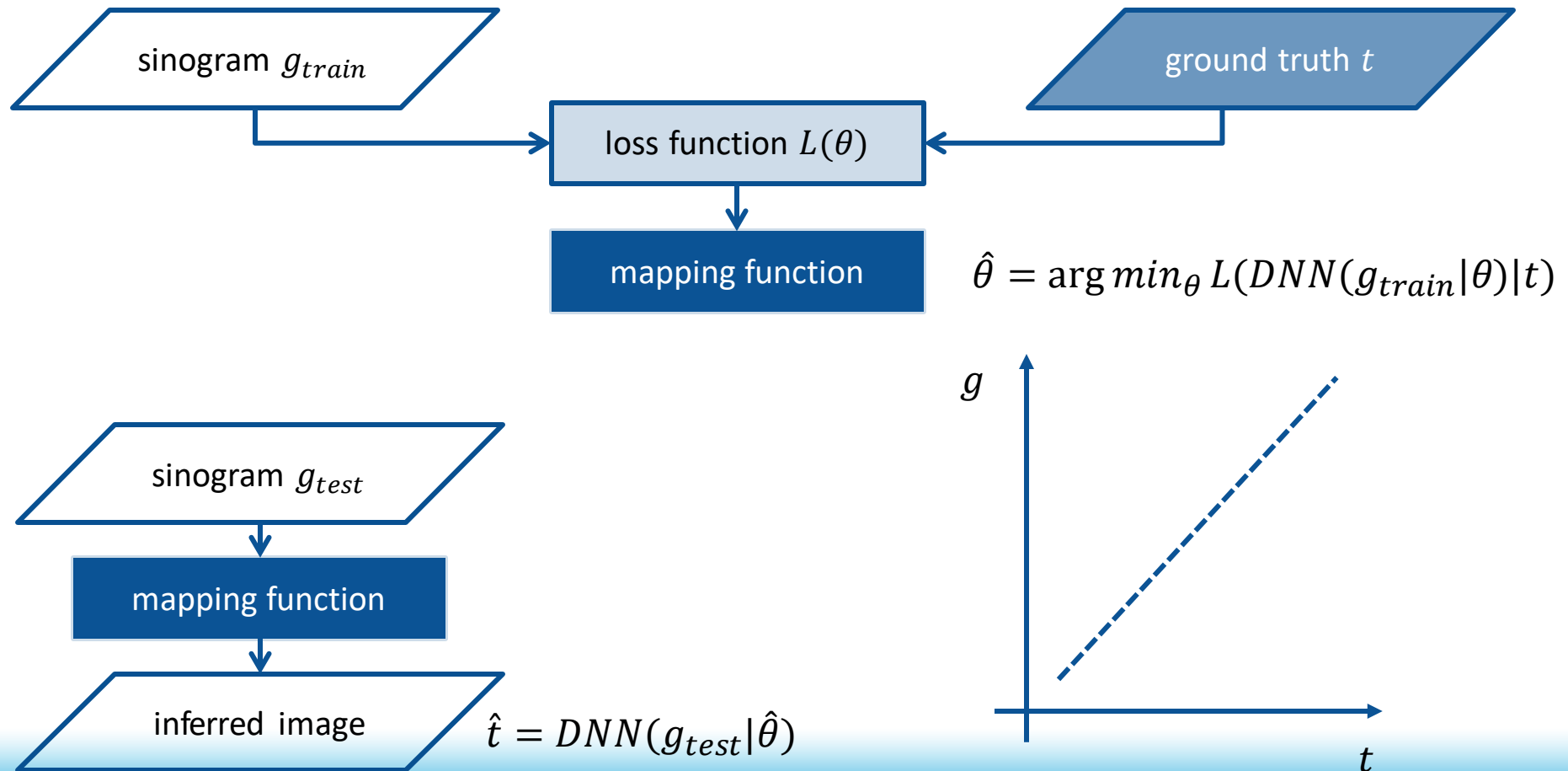
- The **iterative reconstruction paradigm** is to find the image that minimizes the “discrepancy” between the forward-projection of the image (i.e., the **model of the sinogram**) and the acquired sinogram



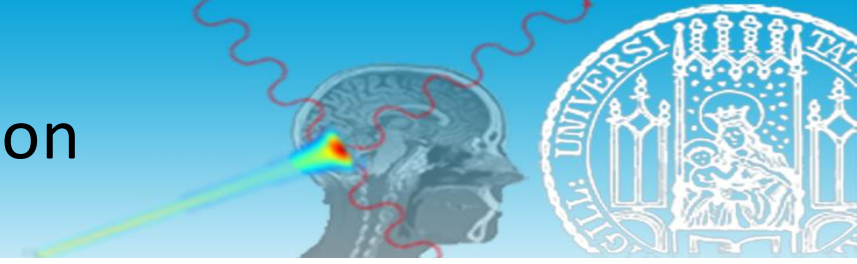
# Deep tomographic reconstruction



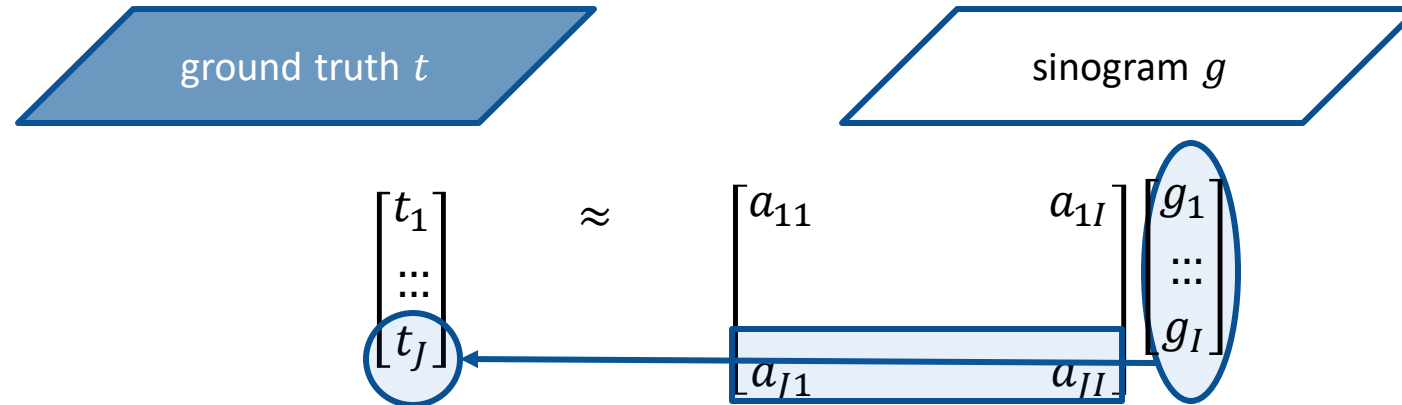
- The machine learning paradigm in tomographic image reconstruction is to find the parameters of the mapping function that infers the ground truth based on supervised prediction



# Deep tomographic reconstruction

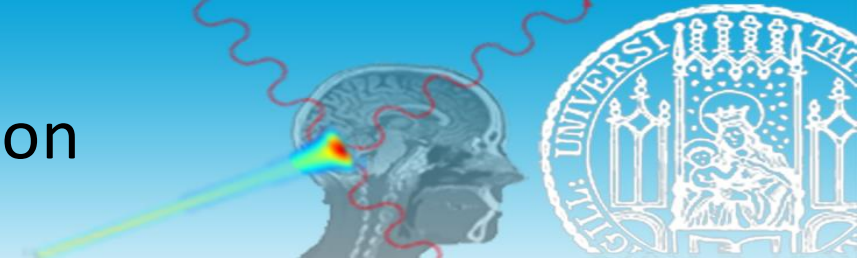


- The back-projection is a linear mapping (i.e., matrix-vector multiplication) that can be described by a **fully connected layer** (i.e., linear layer) of an artificial neural network (ANN)



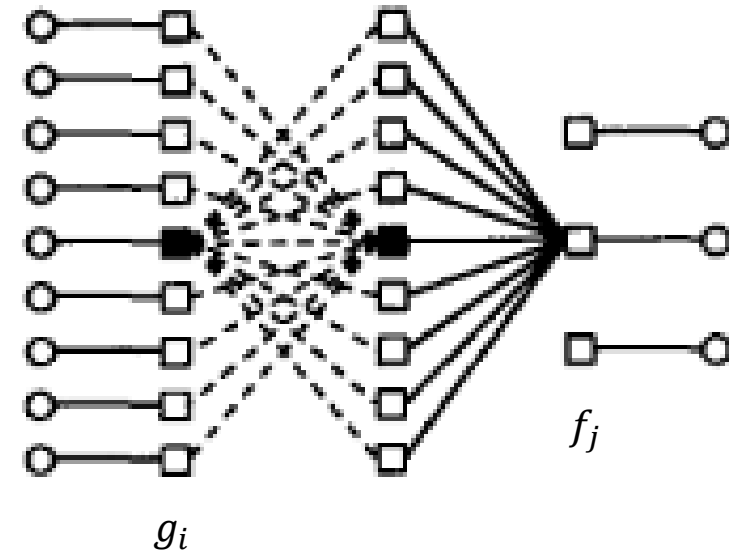


# Deep tomographic reconstruction



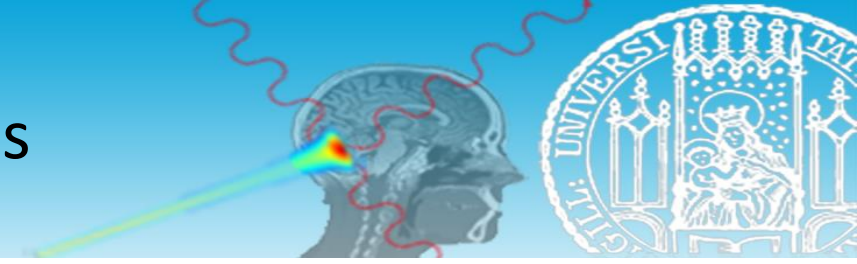
- One of the first ML attempt to deep tomographic reconstruction was based on the “pre-calculation” of the filters for the filtered back-projection, instead of being analytically calculated each time...
  - The learnable weights (learnt based on a point source) are applied along the projection lines of the sinogram
  - The back-projection is implemented for each projection lines of the sinogram as **fully connected layer** with **non-learnable weights**

Floyd, C. E. (1991). An artificial neural network for SPECT image reconstruction. *IEEE transactions on medical imaging*, 10(3), 485-487.

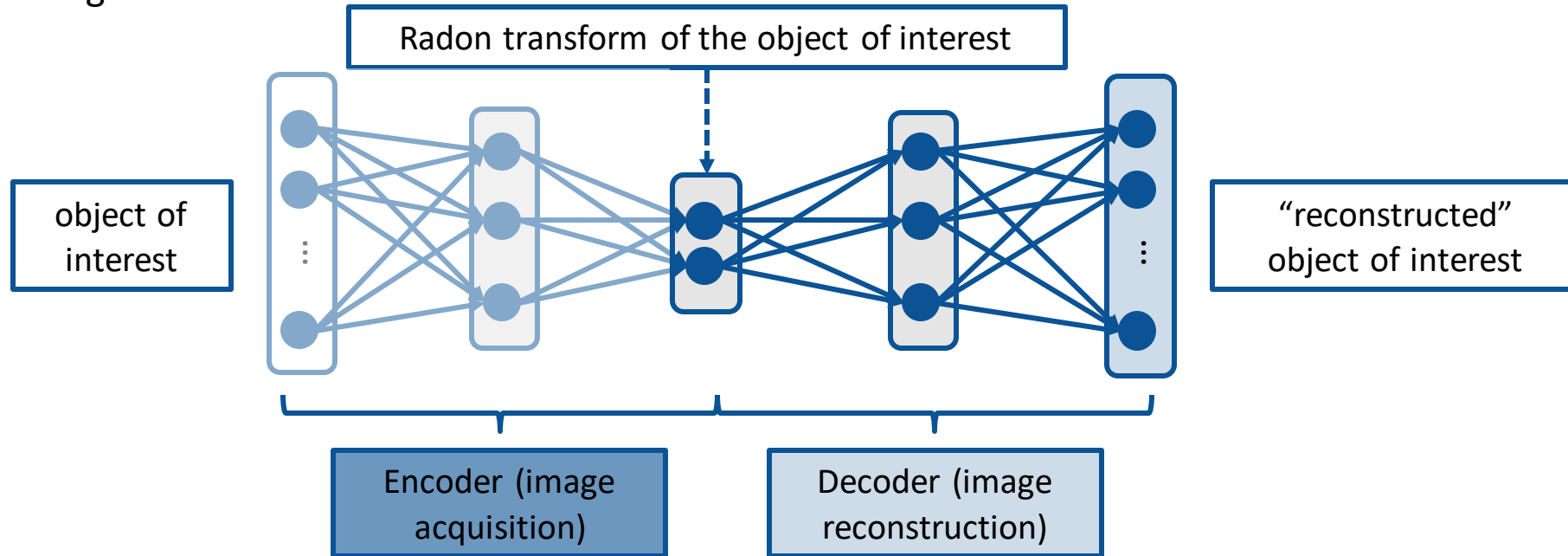


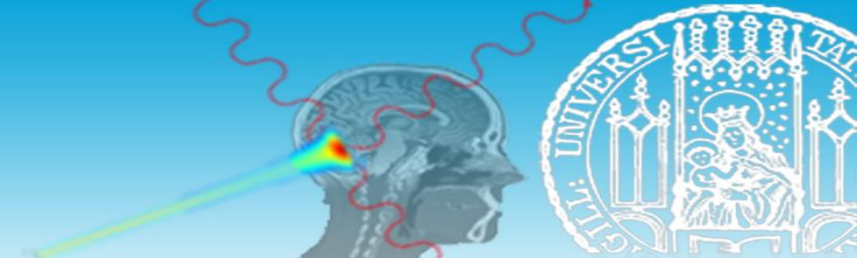
- In practice, this is suitable only for two-dimensional images

# Direct reconstruction methods



- The purpose of domain transform is to map the sinogram (i.e., the projections) to the image
  - The measured sinogram encodes an intermediate representation of the object of interest in the projection domain (i.e., the Radon transform), similar to an encoding function
  - The measured sinogram is subsequently reconstructed into an image by an inversion of the encoding function, similar to a decoding function





- AI is a very capable and potentially very impactful tool to advance medical physics in near future. However, the exact mathematical theory behind is still lacking...
- The **interpretability** is of utmost importance for AI in medical physics
- There are two approaches of reasoning: **deduction** and **induction**. Accordingly, there are two schools of AI

- Deduction is a top-down approach, from knowledge and information towards data, theory-based



- Induction is a bottom-up approach, from data towards information or knowledge, data-driven



- Currently, the mainstream approach of AI is inductive or data-driven. However, it is widely recognized that AI should be capable of being both data-driven and model-based. Towards this goal, networks trained with **big data** could be transformed into **knowledge graphs** so that the unification of data-driven and theory-based learning could be facilitated