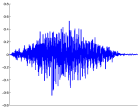
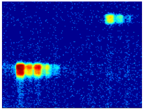
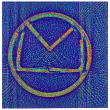
Deep Learning for inverse problems

Daniel Otero, Peter Maass

Center for Industrial Mathematics
University of Bremen

Marseille, 14. 11. 2019

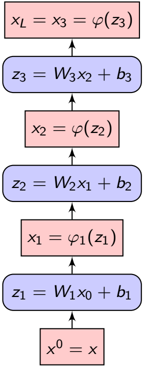



Overview Friday lectures

- Matrix inversion with LISTA
- Invertible networks
- Statistical learning
- Digital pathology
- Magnetic particle imaging MPI

Outline

Formal definition: feedforward neural networks

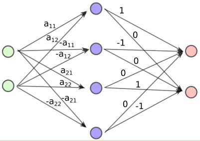


Feedforward neural network with L layers:

$$\begin{aligned} \text{input:}\quad & x_0 \\ & z_k = W_k x_{k-1} + b_k \\ & x_k = \varphi(z_k) \\ \text{ouput:}\quad & x_L \end{aligned}$$

φ non-linear, componentwise, network parameters $\mathcal{W} = \{W_k, b_k\}$
affine transformation, matrix W, bias b

$$\varphi_{\mathcal{W}}(x_0) = x_L = \varphi(W_3\ \varphi(W_2\ \varphi(W_1 x + b_1) + b_2) + b_3)$$



Neural network for 2-by-2 linear system

- Nonlinear activation function φ with $\varphi(x) = \text{ReLU}(x) = \max\{x, 0\}$ (Rectified linear units, applied component-wise)
- $z = a_{11}x_1 + a_{12}x_2 \lessgtr 0$

$$z = ReLU(z) - ReLU(-z)$$

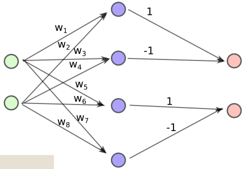
$$w_1 = -w_3 = a_{11}, w_2 = -w_4 = a_{12}, etc.$$

- Employ this net to approximate A_ε (direct problem) or A_ε^{-1} (inverse problem) for

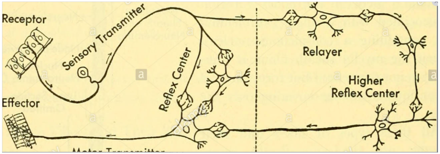
$$A_\varepsilon = \begin{pmatrix} 1 & 1 \\ 1 & 1+\varepsilon \end{pmatrix} \quad \text{for } \varepsilon = 1, 0.1, 0.01, \dots$$

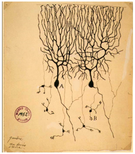
$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

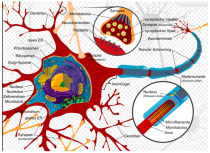
$$\Updownarrow$$



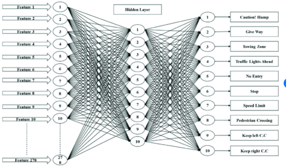
The human neural system



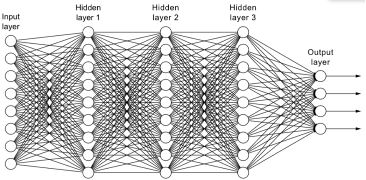




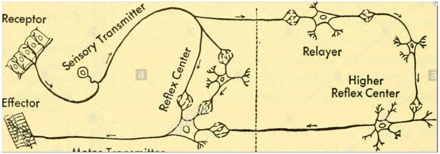
Number of neurons:

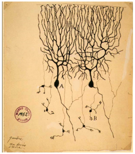


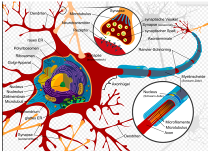
Length of pathways:



The human neural system

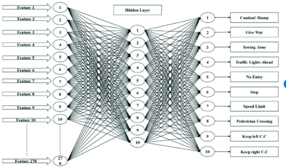


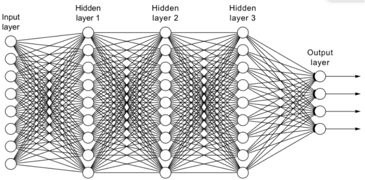




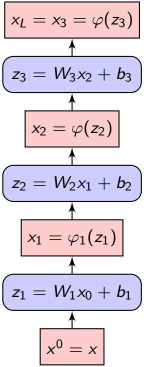
Number of neurons: 80.000.000.000

Length of pathways: 5.000.000 km





Formal definition: feedforward neural networks

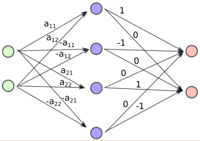


Feedforward neural network with L layers:

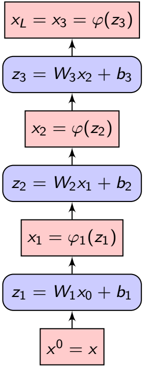
$$\begin{aligned} \text{input:}\quad & x_0 \\ & z_k = W_k x_{k-1} + b_k \\ & x_k = \varphi(z_k) \\ \text{ouput:}\quad & x_L \end{aligned}$$

φ non-linear, componentwise, network parameters $\mathcal{W} = \{W_k, b_k\}$
affine transformation, matrix W, bias b

$$\varphi_{\mathcal{W}}(x_0) = x_L = \varphi(W_3\ \varphi(W_2\ \varphi(W_1 x + b_1) + b_2) + b_3)$$



Feedforward Neural Network

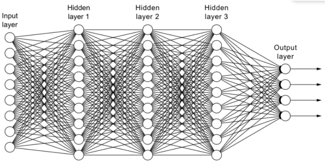


Feedforward neural network with L layers:

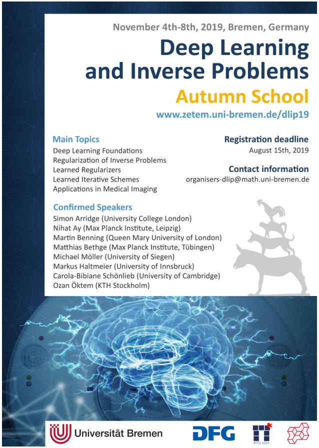
$$\begin{aligned} \text{input:} \quad & x_0 \\ & z_k = W_k x_{k-1} + b_k \\ & x_k = \varphi(z_k) \\ \text{ouput:} \quad & x_L \end{aligned}$$

φ non-linear, componentwise, network parameters $\mathcal{W} = \{W_k, b_k\}$

$$\varphi_{\mathcal{W}}(x_0) = x_L = \varphi(W_3 \; \varphi(W_2 \; \varphi(W_1 x + b_1) + b_2) + b_3)$$



Overview



Overview

Introduction

- Trivial networks
- Inverse problems

DL for inverse problems

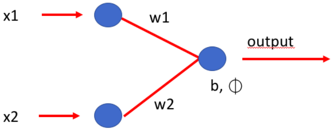
- Unrolling iterations
- Learning with few data

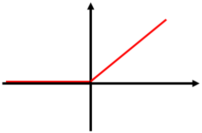
Stochastic approach

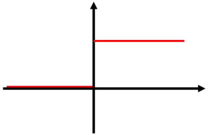
- Bayes approach
- Learning distributions

Trivial networks

- Network with no internal layer, *ReLU*- or threshold nonlinearity
- Multiplication networks, input (x, y), output xy (product)
- Matrix vector multiplication





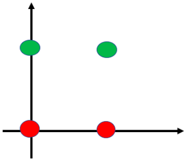


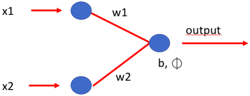
$$\text{output} = \varphi(w1 * x1 + w2 * x2 + b)$$

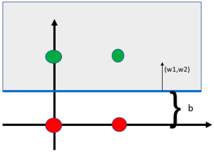
$$w1 * x1 + w2 * x2 + b \ \geq \ 0$$

Trivial networks

- Network with no internal layer, threshold nonlinearity
- Multiplication networks, input (x, y), output xy (product)
- Matrix vector multiplication



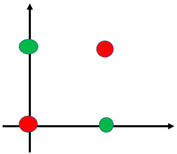


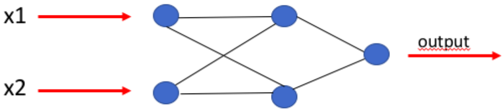


$$\text{output} = \varphi(0 * x1 + 1 * x2 - 0.5)$$

Trivial networks

- XOR-Network with one internal layer, ReLU-non-linearity
- Multiplication networks, input (x, y), output xy (product)
- Matrix vector multiplication

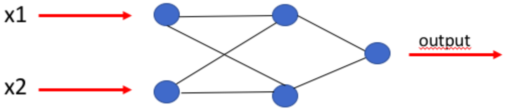




find suitable weights $w_i, i = 1, .., 6,$ and tresholds $b_i, i = 1, .., 3$

Trivial networks

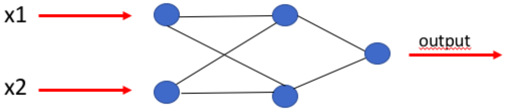
- Network with one internal layer, ReLU-non-linearity
- Multiplication networks, input (x, y), output xy (product)
- Matrix vector multiplication



find suitable weights such that output $= x1 * x2$

Trivial networks

- Network with one internal layer, ReLU-non-linearity
- Multiplication networks, input (x, y), output xy (product)
- Matrix vector multiplication



find suitable weights such that output $= x1 * x2$

precision ε requires $\mathcal{O}(-log(1/\varepsilon))$ nodes,connections

Neural network for 2-by-2 linear system

- Nonlinear activation function φ with $\varphi(x) = \text{ReLU}(x) = \max\{x, 0\}$ (Rectified linear units, applied component-wise)
- $z = a_{11}x_1 + a_{12}x_2 \lessgtr 0$

$$z = ReLU(z) - ReLU(-z)$$

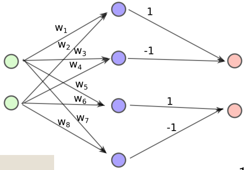
$$w_1 = -w_3 = a_{11}, w_2 = -w_4 = a_{12}, etc.$$

- Employ this net to approximate A_ε (direct problem) or A_ε^{-1} (inverse problem) for

$$A_\varepsilon = \begin{pmatrix} 1 & 1 \\ 1 & 1+\varepsilon \end{pmatrix} \quad \text{for } \varepsilon = 1, 0.1, 0.01, \ldots$$

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

$$\Updownarrow$$



Neural network φ_W

- design/architecture, parameters $\mathcal{W} = (W, b)$
- loss function/discrepancy functional
- algorithm for minimization

Loss function , training data $(x^{(i)}, y^{(i)})$ with $y^{(i)} = Ax^{(i)}$, $i = 1, .., n$

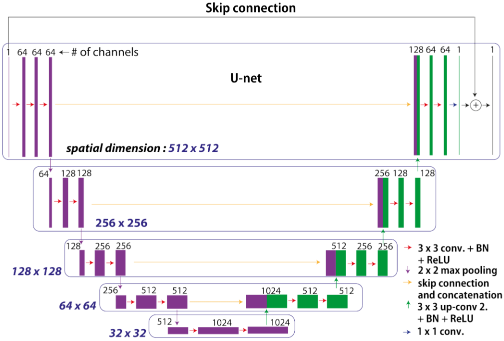
$$L(\mathcal{W}) = \sum_i \|\varphi_{\mathcal{W}}(y^{(i)}) - x^{(i)}\|^2$$

$$W = argmin_W L(\mathcal{W})$$

Application: After training, new data y: $\hat{x} = \varphi_{\mathcal{W}}(y)$

$$\boldsymbol{E}(W) = A \quad , \quad \boldsymbol{E}(\|W - A\|^2) = \mathcal{O}\left(\frac{1}{n}\right) .$$

Convolutional nets: design, training, application



Restriction of parameter set of neural networks

Ronneberger, Olaf; Fischer, Philipp; Brox, Thomas
U-Net: Convolutional Networks for Biomedical Image Segmentation, 2015.

Motivation (Inverse Problems)

- unknown physical quantity
- indirect observation/ measurement
- noisy measurements

Example 2:

determine speed $f(t)$ from measured positions $g(t)$

$$g(t) = f_0 + \int_0^t f(\tau) d\tau$$

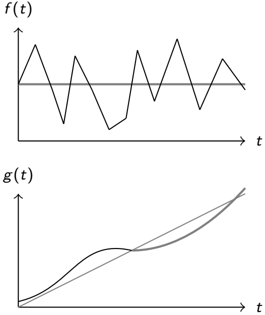
Motivation (Inverse Problems)

- unknown physical quantity
- indirect observation/ measurement
- noisy measurements

Example 2:

determine speed $f(t)$ from measured positions $g(t)$

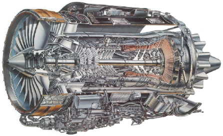
$$g(t) = f_0 + \int_0^t f(\tau) d\tau$$



Motivation (Inverse Problems)

Example 3:

determine internal defects of rotating system from sensor measurements at the surface



- defects/ unbalances of rotating central shaft creates vibrations
- measurements: vibrations on the surface

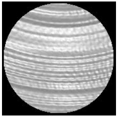
$f(x)$ unbalance at position x

$$M\ddot{u} + D\dot{u} + Su = f$$

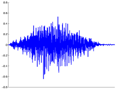
Measure u at sensor positions

Further examples

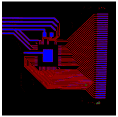
indirect and noisy measurements $\triangleq$ ill-posed reconstruction



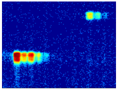
High precision surfaces



Linear guideways



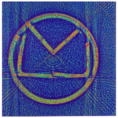
Quality control



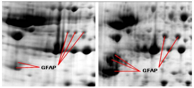
LC-MS spectra

Aero engines

Turning processes



Quality control



Test design

Mathematical Model I

- Unknown f, measurement g, $Af = g$

$$f \xrightarrow[A]{} \mathbf{g}$$

- A transfer operator / system matrix / measurement process
- Discrete model: $f \in \mathbb{R}^n, g \in \mathbb{R}^m$
  - Linear problem A $m \times n$ matrix
  - Non-linear $A : \mathbb{R}^n \to \mathbb{R}^m$ continuous mapping
- Continuous model: $f \in L_2(\Omega)$ or $f \in X$, $g \in Y$, X, Y function spaces

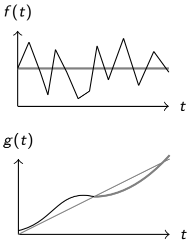
Mathematical Model II, $Af = g$

Inverse problem:
given A, g, determine f

$$g(t) = Af(t) = \int_0^t f(s)ds$$

Common features

- A strongly damping, large deviations of input f causes only small deviations of output g



Example: speed $f_1(t) = 1$,
$f_2(t) = 1 + \partial f(t)$
$\partial f(t)$ highly oscillating
exact position $g_1(t) = t, \quad g_2(t) \approx t.$

Mathematical Model III, $Af = g$

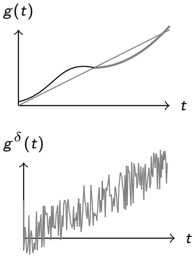
Inverse problem: given A, g, determine f

Common features

- limited measurement precision/ measurement errors
- available data $g^\delta = g + \eta$, η random noise

Example: measured position

$$g_1^\delta(t) = g_1(t) + \eta_1,$$
$$g_2^\delta(t) = g_2(t) + \eta_2$$

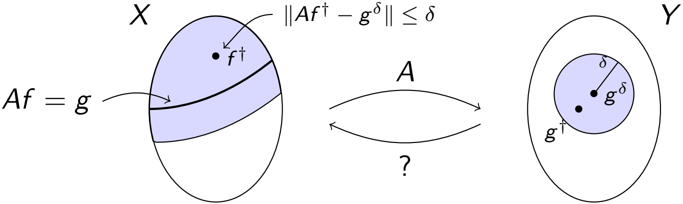


How to determine approximations to f_1, f_2 from these noisy measurements?

Mathematical Model IV

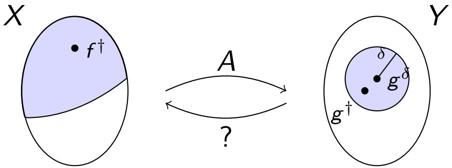
$$A : X \to Y, \quad X, Y \text{ function spaces}$$

- $f^\dagger \in X$ unknown physical quantity
- $g^\dagger = Af^\dagger$ exact data
- $g^\delta = g^\dagger + \eta$ available noisy data $\|g - g^\delta\| \leq \delta$
- "solutions" $\{f \mid \|Af - g^\delta\| \leq \delta\}$ unbounded set



Regularization Theory

"Determine an approximation to the solution of $Af^\dagger = g^\dagger$ from measured noisy data g^δ"



"solution set" $\{f \mid \|Af - g^\delta\| \le \delta\}$

stabilization/ regularization: Tikhonov functionals
Iteration methods (Landweber, CG)
Truncated singular value decomposition

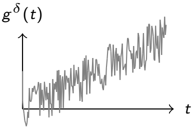
Tikhonov Functionals

- Compute approximation by minimizing

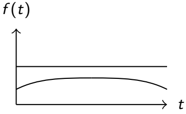
$$J_\alpha(f) := \|Af - g^\delta\|^2 + \alpha\|f\|^2$$

- Notation: $f_\alpha^\delta := \operatorname{argmin} J_\alpha$
- A linear operator: $f_\alpha^\delta = (A^*A + \alpha I)^{-1} A^* g^\delta$
- A^{-1} unbounded, $\|(A^*A + \alpha I)^{-1}\| \leq \frac{1}{\alpha}$

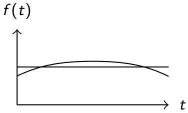
Example:



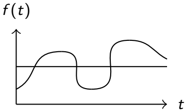
measurement data



α too large



optimal α



α too small

Tikhonov Functionals (Theory)

$$\min \|Af - g^\delta\|^2 + \alpha\|f\|^2$$

$$\min \quad dist(Af, g^\delta) + R_\alpha(f)$$

- Choice of regularization parameter $\alpha = \alpha(\delta, g^\delta)$
- Convergence properties $f_\alpha^\delta \stackrel{?}{\longrightarrow} f^\dagger$ for $\delta \to 0$
- Numerical schemes for computing f_α^δ

Theorem. If $\alpha \approx \delta^{1/2}$ and if $f^\dagger \in \text{range}(A^*)$, then

$$\|f_\alpha^\delta - f^\dagger\| = \mathcal{O}(\delta^{1/2}).$$

Remarks: Morozov discrepancy principle, source conditions, QR-decomposition, CG algorithm

Tikhonov Functionals Non-linear Operator

$$\min \|A(f) - g^\delta\|^2 + \alpha \|f\|^2$$

Theorem. (Engl et al. 1989) If A has Lipschitz continuous Frechet derivative and if $f^\dagger \in \text{range}(A'(f^\dagger))$, then $\alpha \approx \delta^{1/2}$ yields

$$\|f_\alpha^\delta - f^\dagger\| = \mathcal{O}(\delta^{1/2}).$$

Iteration methods:

Gradient descent for discrepancy term $\|A(f) - g^\delta\|^2$

$$f^{k+1} = f^k - \lambda\, A'(f^k)^* \left(A(f^k) - g^\delta\right)$$

Accelerated iteration methods, dual iteration

Inverse problems vs. control theory

$$\min \|A(f) - g\|^2 + R_\alpha(f)$$

A known system operator, g given output, determine input f

$f^\dagger$ unknown physical quantity	control parameter
existence "clear"	attainability, controlability
g measured, noisy	desired state, artificially designed
convergence $\|f_\alpha^\delta - f^\dagger\| \to_{\delta\to 0} 0$	output $\|A(f) - g\| \to 0$

Control theory $\Longleftrightarrow$ Inverse problems
Uni Graz (K. Kunisch), University College London (B. Jin, S. Arridge), WIAS (M. Hintermüller), G. Uhlmann (MIT), Y. Nesterov (Louvain), ...