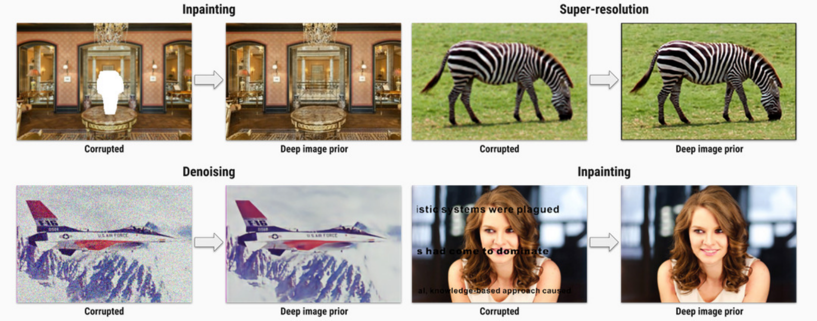
EARLY STOPPING OF UNTRAINED NEURAL NETWORKS

Tim Jahn (joint with Bangti Jin)

London, 23.5.23

NEURAL NETWORKS IN IMAGING SCIENCE

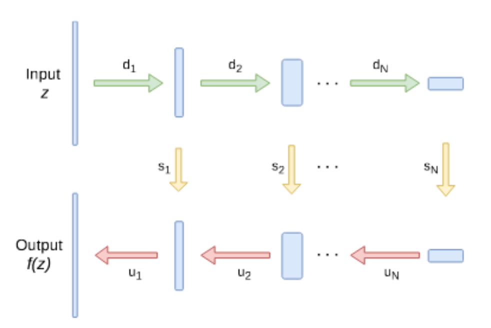


Supervised learning: Network's weights are adjusted through training on paired data

all images from **Ulyanov et al. 2018**

UNTRAINED NEURAL NETWORKS

Popular network architecture for image tasks: **U-net**

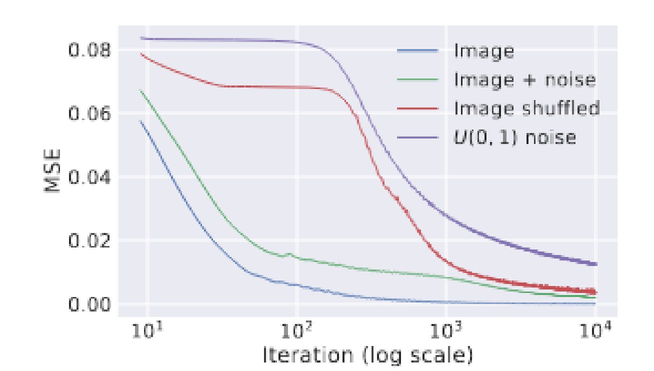


What if no training data available?

Deep image prior (Ulyanov et al. 2018)
Network weight's are tuned to fit a single image from random input

Uljanov et al. use highly over-parametrised U-net.

network can fit any output, but natural images substantially faster



„Regularisation by architecture"+ early stopping

ILL-POSED INVERSE PROBLEMS

(Discrete) inverse problem:

$$Ax = y^\delta$$

- forward model $A \in \mathbb{R}^{m \times n}$,
- noisy data $y^\delta = y^\dagger + \delta\xi \in \mathbb{R}^n$,
- exact solution $x^\dagger = A^+ y^\dagger$.

Problem: A ill-conditioned $\Rightarrow$ standard inversion does not work.

Example: Computerised tomography, image deblurring

OUR SETTING

We use

$$G(C) := \mathrm{ReLU}(UC)v$$

to approximate $x^{\dagger}$.

- $\mathrm{ReLU} = \mathrm{ReLU}(s) = \max(s, 0)$ rectifier linear unit applied componentwise,
- $U \in \mathbb{R}^{n \times n}$ (usually a convolution)
- $v \in \mathbb{R}^{N}$ normalised with entries ± 1,
- $C \in \mathbb{R}^{n \times N}$ **weights** to be tuned

TUNING OF WEIGHTS C

Apply gradient descent to

$$\mathcal{L}(C) := \frac{1}{2}\|AG(C) - y^\delta\|^2$$

with random Gaussian initilisation C_0.

Discrepancy principle for stopping of the iteration:

$$k_{\mathrm{dp}}^\delta := \min\left\{k \geq 0 \;:\; \|AG(C_k) - y^\delta\| \leq \tau\delta\right\}$$

OPTIMAL CONVERGENCE

Source condition: $\mathcal{X}_{\nu,\rho} := \left\{x \in \mathbb{R}^n \;:\; x = (A^T A)^{\nu/2} w,\; \|w\| \leq \rho\right\}$

worst-case-error rate: $\mathrm{err}_{WC}(\delta, \rho, \nu)$

THEOREM (J., JIN)

Assume that A and $\Sigma(U)$ have polynomially decaying singular values and aligned right singular vectors. Then, for $N = N(\delta, \epsilon)$ large enough and $\omega = \omega(\delta, \epsilon)$ small enough and a constant $C > 0$, it holds that

$$\inf_{x^\dagger \in \mathcal{X}_{\rho,\nu}} \mathbb{P}\left(\|G(C_{k^\delta_{\mathrm{dp}}}) - x^\dagger\| \leq C\, \mathrm{err}_{WC}(\delta, \rho, \nu) \right) \geq 1 - \epsilon.$$

„Optimal convergence for large enough network"

$\mathcal{J}(C_0)$ jacobian at random initialisation C_0.

$$\mathbb{E}\left[\mathcal{J}(C_0)\mathcal{J}(C_0)^T\right] = \left(\frac{1}{2}\left(1 - \frac{1}{\pi}\cos^{-1}\left(\frac{(u_i, u_j)}{\|u_i\|\|u_j\|}\right)\right)(u_i, u_j)\right)_{i,j=1}^{n}$$
$$=: \Sigma(U) = JJ^T$$

J reference jacobian

$\implies$ Jacobian at initialisation approximately only dependent on U

Compare dynamics of nonlinear and linear least squares

$$\mathcal{L}(C) = \frac{1}{2}\|AG(C) - y^{\delta}\|$$

,

$$\mathcal{L}^{\mathrm{lin}}(C) = \frac{1}{2}\|AG(C_0) + AJ(c - c_0) - y^{\delta}\|^2$$

(c, c_0 are vectorised versions of C, C_0)

$\Longrightarrow$ For not too many iterations, linear and nonlinear iterates (and residuals) stay close

ERROR DECOMPOSITION

- C_k nonlinear iterates, C_k^{lin} linear iterates,
- $G^{\text{lin}}(\cdot) = G(C_0) + J(\cdot - c_0)$ linearised network,

$$\|G(C_k) - x^\dagger\| \leq \|G(C_k) - G^{\text{lin}}(C_k^{\text{lin}})\| + \|G^{\text{lin}}(C_k^{\text{lin}}) - x^\dagger\|$$

First term:

$$\begin{aligned}\|G(C_k) - G^{\text{lin}}(C_k^{\text{lin}})\| &\leq \|G(C_k) - G^{\text{lin}}(C_k)\| + \|G^{\text{lin}}(C_k) - G^{\text{lin}}(C_k^{\text{lin}})\| \\ &\leq \sup_{\xi \in \text{conv}(c_k, c_0)} \|\mathcal{J}(\xi) - J\| \|c_k - c_0\| + \|J(c_k^{\text{lin}} - c_k)\| \\ &\leq \text{small}\end{aligned}$$

for k not too large (dependent on N and ω).

For remainder $\|G^{\mathrm{lin}}(C^{\mathrm{lin}}_{k^{\delta}_{\mathrm{dp}}}) - x^{\dagger}\|$ mainly classical analysis.

Two issues:

- representation of x through linearised network
- random initialisation

$\Longrightarrow$ Optimal rates for suitable aligned sinuglar vectors of A and $\Sigma(U)$ and polynomially decaying singular values.

Conclusion

- untrained neural networks provably can be used to solve inverse problems
- early-stopping essential
- discrepancy principle realises early stopping

OUTLOOK

Problem: N „extremely“large (e.g., $N \geq n \log(\epsilon^{-1})\delta^{-21}$)

One approach: Pretraining on simpler data sets, in order to identify relevant subspaces.

Example: Radon transform

- application: x-ray tomography for medical diagnosis
- pretraining on synthetic data (ellipsoids,...)
