Statistical analysis of an image classification problem

Johannes Schmidt-Hieber

joint work with Sophie Langer

motivation

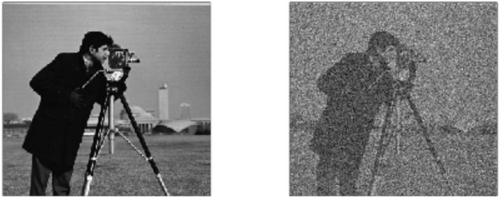
- statistical theory for deep networks focused on nonparametric regression model, that is, data $(\mathbf{X}_i, Y_i)$, $i = 1, \ldots, n$ satisfy

$$Y_i = f(\mathbf{X}_i) + \varepsilon_i$$

- ε_i is the measurement noise
- nonparametric regression is very well understood
- previous work shows that empirical risk minimizer taken over suitable classes of deep ReLU networks achieve minimax estimation rates

image classification vs. nonparametric regression

sample images for image classification (Krishevsky et al. 2012)



nonparametric regression is a denoising problem

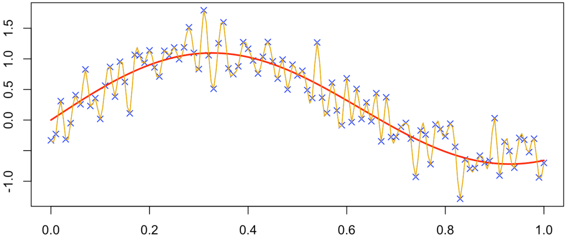
theory of deep learning

Goal: Theoretical underpinning for new phenomena observed in DL

- good approximation properties ✓
- exploit low-dimensional structure in the data ✓
- fast convergence rates ✓
- outperforms other methods for complex tasks ✓
- overparametrization ✗
- zero training error ✗
- data augmentation ✗
- ...

for data denoising, fitting overparametrized networks, zero training error and data augmentation are detrimental

overparametrization



- for overparametrized shallow ReLU network with properly chosen learning rate, SGD converges to natural cubic spline interpolant
- $\rightsquigarrow$ inconsistent estimator

prediction

- in deep learning we are ultimately interested in prediction
- suppose that $\widehat{Y}$ is a predictor based on $(\mathbf{X}_i, Y_i)$, $i = 1, \ldots, n$ from nonparametric regression model

$$Y_i = f(\mathbf{X}_i) + \varepsilon_i$$

- given a new $\mathbf{X}$ from a pair $(\mathbf{X}, Y)$, prediction error is

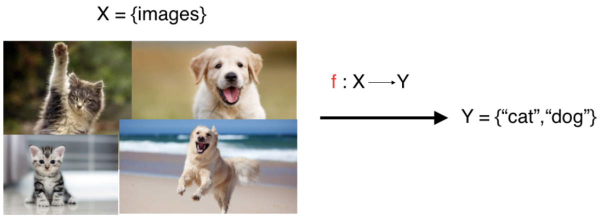
$$E\left[(\widehat{Y} - Y)^2\right] = \mathrm{Var}(\varepsilon_i) + E\left[(\widehat{Y} - f(\mathbf{X}))^2\right]$$

- noise level $\mathrm{Var}(\varepsilon_i)$ dominates prediction error (unless $\mathrm{Var}(\varepsilon_i) \to 0$ quickly enough)

models for image classification

previous discussion motivates to introduce and analyze statistical models describing image classification

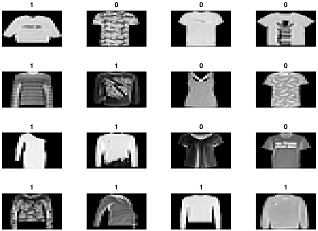
what is the dimension of the problem ?



Machine learning perspective

- (dimension = input dimension) each pixel is a variable and we learn a d-dimensional function
- MNIST dimension $28 \times 28 = 784$
- curse of dimensionality $\rightsquigarrow$ problem is considerably harder for larger images

what is the dimension of the problem ?



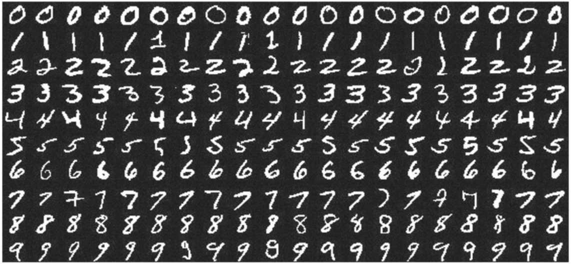
Data modelling:

- dimension $= 2$: view pixelated image as a matrix $\mathbf{X} = (X_{j,\ell})_{j,\ell=1,\dots,d}$ with

$$X_{j,\ell} = f\left(\frac{j}{d}, \frac{\ell}{d}\right)$$

- unknown function $f : [0,1]^2 \to [0,\infty)$
- $\rightsquigarrow$ increasing the number of pixels leads to higher image resolution and therefore a better performance

how to model random image deformations?



classification:

- every image in one class is a random deformation of a template image
- how to model random deformations?
  - functional data analysis
  - image registration
- here we propose a very simple model

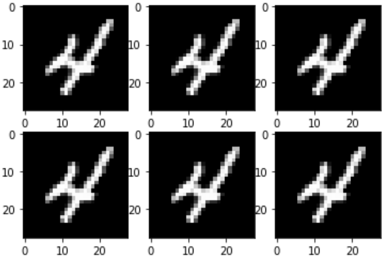
a simple image deformation model

Image $\mathbf{X} = (X_{j,\ell})_{j,\ell=1,\ldots,d}$ with

$$X_{j,\ell} = f\left(\frac{j}{d}, \frac{\ell}{d}\right)$$

and

- template function $f : \mathbb{R}^2 \to [0, \infty)$



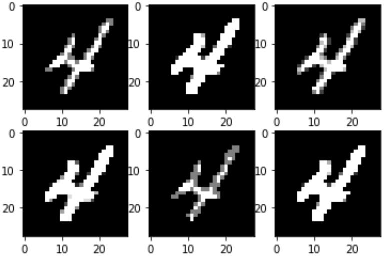
a simple image deformation model

Image $\mathbf{X} = (X_{j,\ell})_{j,\ell=1,\ldots,d}$ with

$$X_{j,\ell} = \eta f\left(\frac{j}{d}, \frac{\ell}{d}\right)$$

and

- template function $f : \mathbb{R}^2 \to [0, \infty)$
- illumination factor η



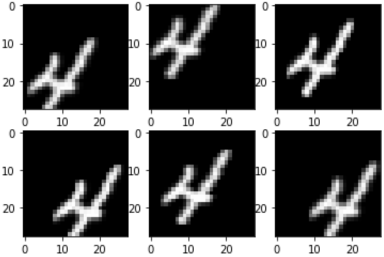
a simple image deformation model

Image $\mathbf{X} = (X_{j,\ell})_{j,\ell=1,\ldots,d}$ with

$$X_{j,\ell} = f\left(\frac{j}{d} - \tau, \frac{\ell}{d} - \tau'\right)$$

and

- template function $f : \mathbb{R}^2 \to [0, \infty)$
- shifts τ, τ'



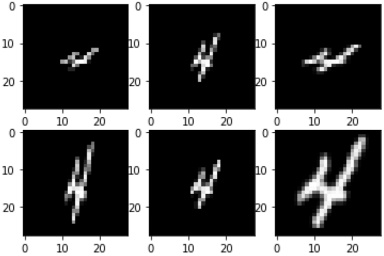
a simple image deformation model

Image $\mathbf{X} = (X_{j,\ell})_{j,\ell=1,\ldots,d}$ with

$$X_{j,\ell} = f\left(\xi\frac{j}{d}, \xi'\frac{\ell}{d}\right)$$

and

- template function $f : \mathbb{R}^2 \to [0,\infty)$
- scaling ξ, ξ'



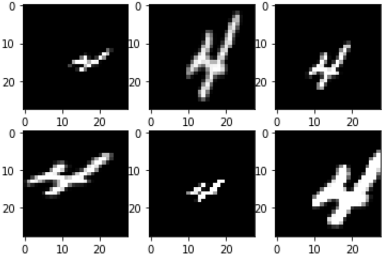
a simple image deformation model

Image $\mathbf{X} = (X_{j,\ell})_{j,\ell=1,\dots,d}$ with

$$X_{j,\ell} = \eta f\left(\xi \frac{j}{d} - \tau, \xi' \frac{\ell}{d} - \tau'\right)$$

and

- template function $f : \mathbb{R}^2 \to [0, \infty)$
- illumination factor η
- shifts τ, τ'
- scaling ξ, ξ'



binary image classification

Given:

n pairs $(\mathbf{X}_i, k_i) \in [0,\infty)^{d\times d} \times \{0,1\}$ with $\mathbf{X}_i = (X_{j,\ell}^{(i)})_{j,\ell=1,\ldots,d}$ and

$$X_{j,\ell}^{(i)} = \eta_i f_{k_i}\left(\xi_i \frac{j}{d} - \tau_i, \xi_i' \frac{\ell}{d} - \tau_i'\right),$$

where

- f_0, f_1 are unknown
- $(\eta_i, \xi_i, \xi_i', \tau_i, \tau_i')$ are unobserved i.i.d. random vectors
- we assume that the object is fully visible on the image $\rightsquigarrow$ constraints on support of f_0, f_1 and $(\eta_i, \xi_i, \xi_i', \tau_i, \tau_i')$
- zero background

while in denoising problems, methods have to learn local smoothing, a learning method applied to the above data will need to **learn invariance** of the class label under the possible transformations!

two approaches

we analyze:

- classification via image alignment: new method specifically designed for this data model
  - what is optimal?
- CNNs
  - how well can CNNs learn underlying invariance?

for both methods we obtain bounds for the misclassification error

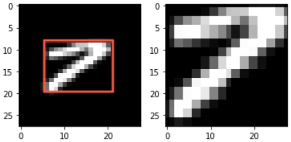
the classifier

steps: Given new image $\mathbf{X}$

(i) find the rectangular support

(ii) rescale image such that rectangular support becomes $[0, 1]^2$
$\rightsquigarrow$ (near) independence on shifts, scaling

(iii) normalize brightness
$\rightsquigarrow$ independence on brightness



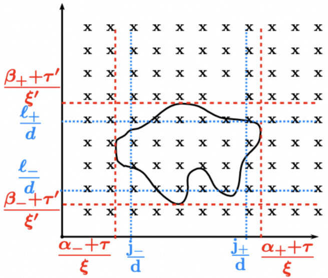
rescaling of rectangular support

For $T_{\mathbf{X}}$ the transformed image, we consider one-nearest neighbor classifier

$$\hat{i} \in \arg\min_{i=1,\ldots,n} \| T_{\mathbf{X}} - T_{\mathbf{X}_i} \|_2, \quad \hat{k} := k_{\hat{i}}$$

- interpolating classifier

rectangular support



support (black), true rectangular support, measured rectangular support

- main source of error is the discretization error occurring through the rectangular support
- to control the error, we impose a regularity assumption on the support

main result

Theorem: Suppose

(i) labels 0 and 1 occur at least once in the training data

(ii) template functions f_0, f_1 are Lipschitz, support satisfies regularity condition

(iii) minimal separation condition

$$\inf_{a,b,b',c,c' \in \mathbb{R}} \|af_0(b \cdot + c, b' \cdot + c') - f_1\|_2 \gtrsim \frac{1}{d},$$

(recall: images are $d \times d$)

Then classifier perfectly recovers the label

$$k = \hat{k}.$$

lower bound

Theorem: Under the same assumptions, there exist non-negative Lipschitz continuous functions f_0, f_1 with

$$\left\| \eta f_0(\xi \cdot + \tau, \xi' \cdot + \tau') - f_1 \right\|_2 \geq \frac{1}{8d},$$

such that the data generating model can be written as

$$X_{j,\ell} = f_1\left(\frac{j}{d}, \frac{\ell}{d}\right) = \eta f_0\left(\xi \frac{j}{d} + \tau, \xi' \frac{\ell}{d} + \tau'\right).$$

convolutional neural networks

- previous estimator was very much adapted to the specific model
- can CNNs also do perfect classification under the optimal separation condition?

convolutional neural networks

- three components: Convolutional, pooling and fully connected layers
- **convolution**: Slide over the image spatially, computing convolutions
- objective: Extract high-level features
- each convolutional layer contains a series of filters
- finally an activation function is applied to these filters

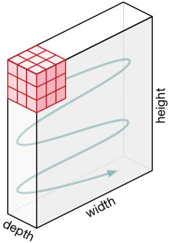


Figure: *

Source:https://towardsdatascience.com/

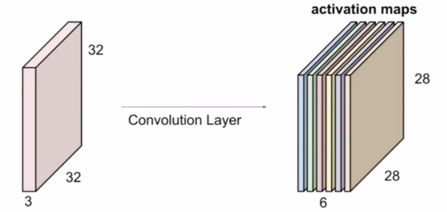


Figure: *

Source:http://cs231n.stanford.edu/

- one convolutional layer with ReLU activation function $\sigma(x) = \max\{x, 0\}$
- k feature maps with filters $\mathbf{W}_1, \ldots, \mathbf{W}_k$
- one global max-pooling layer

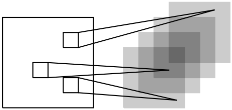


Figure: Illustration of a convolutional layer

The s-th feature map ($s \in \{1, \ldots, k\}$) can be described by

$$\mathbf{o}_s = \sigma(\mathbf{W}_s \star \mathbf{X})$$

and

$$f_{\mathbf{w}}(\mathbf{X}) = (|\mathbf{o}_1|_\infty, \ldots, |\mathbf{o}_k|_\infty).$$

Deep ReLU network function with L hidden layers and width vector $\mathbf{k} = (k_0, \ldots, k_{L+1})$

$$\mathbf{x} \mapsto f(\mathbf{x}) = \Phi_\beta W_L \sigma_{\mathbf{v}_L} W_{L-1} \sigma_{\mathbf{v}_{L-1}} \cdots W_1 \sigma_{\mathbf{v}_1} W_0 \mathbf{x}$$

with softmax function

$$\Phi_\beta(x_1, x_2) = \left(\frac{e^{\beta x_1}}{e^{\beta x_1} + e^{\beta x_2}}, \frac{e^{\beta x_2}}{e^{\beta x_1} + e^{\beta x_2}} \right)$$

and learnable parameters

- $k_i \times k_{i+1}$ matrices W_i
- k_i-dimensional vectors $\mathbf{v}_i$

our CNN architecture

consider CNN class $\mathcal{G}(m)$ defined by

- one convolutional layer with $2m$ convolutional filters
- afterwards max-pooling
- afterwards $L_m = 1 + 2\lceil \log_2 m \rceil$ fully connected layers of width $4m$
- two outputs (binary classification)

invariance of CNNs

- CNNs are (nearly) invariant under shifts
- but not under different rescaling of the object
- it is also known that CNNs have problems to learn scale invariance
- $\rightsquigarrow$ this has sparked some work on scale-invariant CNNs and whether this is desirable

Supervised learning framework with i.i.d. $(\mathbf{X}_1, k_1), \ldots, (\mathbf{X}_n, k_n)$

- k_i is the i-th label $\in \{0, 1\}$
- classes do not need to be balanced
- as pre-processing step, $\mathbf{X}_i$ are normalized

$$\bar{\mathbf{X}}_i := \frac{1}{\|\mathbf{X}_i\|_2}\mathbf{X}_i$$

- least squares loss over CNN class $\mathcal{G}(m)$

$$\widehat{\mathbf{p}} \in \operatorname*{arg\,min}_{\mathbf{p}\in\mathcal{G}(m)} \frac{1}{n}\sum_{i=1}^{n} \left\|\mathbf{Y}_i - \mathbf{p}(\overline{\mathbf{X}_\mathbf{i}})\right\|_2^2$$

$$= \operatorname*{arg\,min}_{\mathbf{p}=(p_0,p_1)\in\mathcal{G}(m)} \frac{2}{n}\sum_{i=1}^{n} \left(k_i - p_1(\overline{\mathbf{X}}_i)\right)^2$$

with $\mathbf{Y}_i = (1 - k_i, k_i)$

$\rightsquigarrow$ provides an estimator for conditional class probabilities

$$p_j(\mathbf{x}) := \mathbf{P}(k = j | \mathbf{X} = \mathbf{x}), \quad j = 1, 2$$

classifier

- least squares fit returns the network $\widehat{\mathbf{p}} = (\widehat{p}_0, \widehat{p}_1)$
- given new image $\mathbf{X}$, the classifier is

$$\widehat{k}(\mathbf{X}) = \mathbf{1}\big(\widehat{p}_1(\mathbf{X}) \geq 1/2\big)$$

statistical risk bound

Theorem: Suppose

- object is fully visible
- template functions f_0, f_1 are Lipschitz
- consider CNN classifier $\widehat{k}(\mathbf{X})$ constructed as above with $\widehat{\mathbf{p}} \in \mathcal{G}_{\Phi_\beta}(m)$ for suitable $m \asymp d^2$
- $\beta = d^2$
- separation criterion

$$\inf_{a,b,b',c,c' \in \mathbb{R}} \|af_0(b \cdot +c, b' \cdot +c') - f_1\|_2 \gtrsim \frac{1}{\sqrt{d}}.$$

Then misclassification error is bounded

$$\mathbf{P}\big(\widehat{k}(\mathbf{X}) \neq k\big) \lesssim d^2 \sqrt{\frac{\log(n) \log^3(d)}{n}} + e^{-d}.$$

some comments on the rate

- for $d, n \to \infty$ and $n \gg d^4 \log^4 d$ the missclassification error converges to zero
  - rate can very likely be improved
- most previous results do not satisfy that
- interesting underlying approximation theory
  - separate filters in convolutional layer for different scales
  - fully connected layers implement maximum function
- minimum separation of order $1/\sqrt{d}$ compared to $1/d$ for the first method
- no condition on support assumed
- data augmentation could reduce sample complexity
  $\rightsquigarrow$ Dependency among training data requires new statistical framework

- show that $\min_{q:[0,1]^2 \to \{0,1\}} \mathbf{P}(q(\mathbf{X}) \neq k(\mathbf{X})) = 0$
- Use

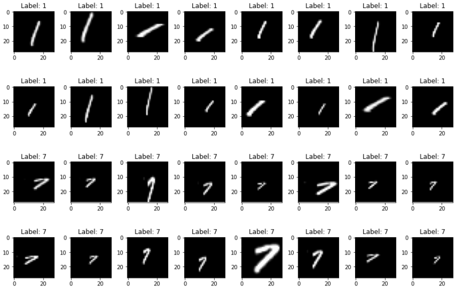
$$\mathbf{P}\big(\widehat{k}(\mathbf{X}) \neq k(\mathbf{X})\big) \leq 2\sqrt{\int \big(\widehat{p}_2(\mathbf{x}) - p(\mathbf{x})\big)^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x})}.$$

- $\rightsquigarrow$ oracle inequalities decompose error in approximation error and sample complexity term
- bound approximation error by $|\widehat{\mathbf{p}}(\mathbf{X}) - \mathbf{Y}|_\infty \leq e^{-d}$
- bound complexity of the network class by VC dimension

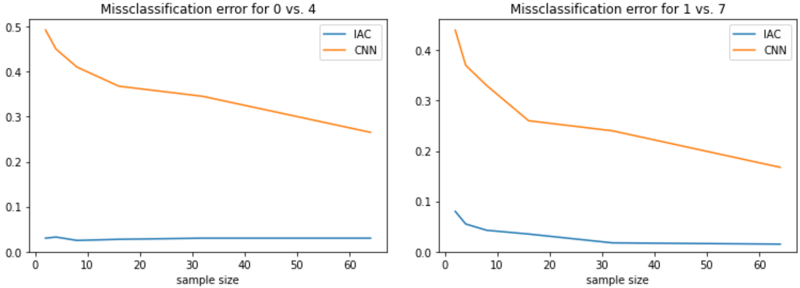
estimators on the MNIST data

Setting

- MNIST dataset: 60.000 examples of handwritten digits from 0 to 9 $\rightsquigarrow d = 28$
- binary classification: pick two classes
- choose *one* image of each class and apply random deformation
- sample size $n \in \{2, 4, 8, 16, 32, 64\}$



empirical results



possible extensions

- background noise
- multiple objects
- other transformations, in particular rotations
- replace constant shifts τ, τ' by functions $\tau(x, y), \tau'(x, y)$
  $\rightsquigarrow$ describes local deformations ($\nearrow$ work by Mallat)
- ODE models for random image deformations proposed in image registration literature
  - generate random vector field u,
  - $\mathbf{X}$ template image
  - randomly deformed image is $\mathbf{X}(1)$, where

$$\partial_t \mathbf{X}(t) = u(\mathbf{X}(t)) \quad \text{with } \mathbf{X}(0) = \mathbf{X}.$$

conclusion

- study deep learning for random image deformation model
- derived bound on misclassification error for CNNs
- many open problems and various extensions are possible

preprint: arXiv:2206.02151

Thank you for your attention!