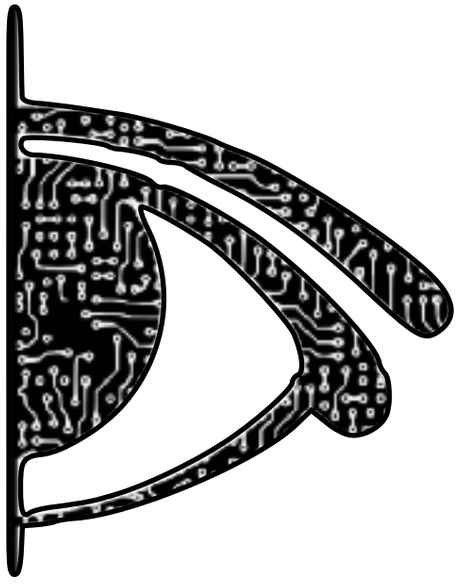
Theoretical results (or problems) in Deep Learning

Edouard Oyallon



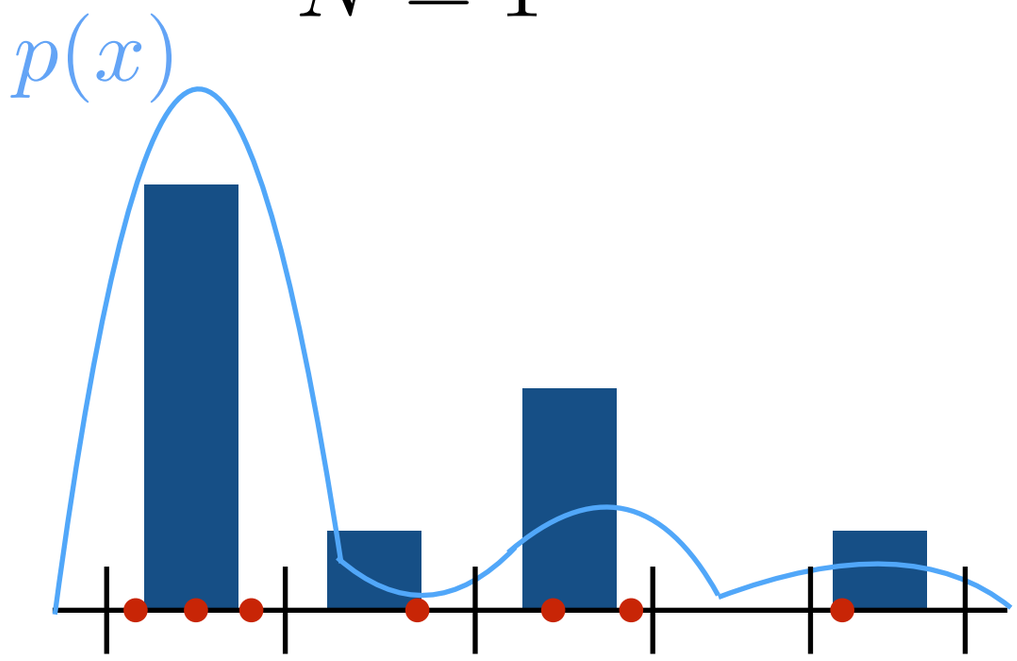
Outline

- High dimensional problems
- Properties of CNNs intermediary representations
- Generalization results
- Optimisation
- Architecture
- Analysis of the CNNs operators

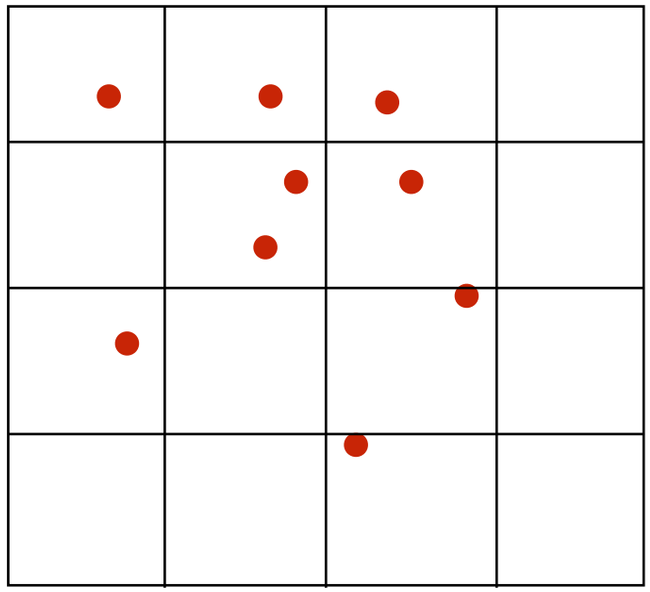
High dimensional issues, an example

- PdFs are difficult to estimate in high dimension.

$N = 1$



$N = 2$



- For a fixed number of points and bin size, as N increases, the bins are likely to be empty.

Curse of dimensionality:
occurs in many machine learning problems

High Dimensional classification

$$(x_i, y_i) \in \mathbb{R}^{224^2} \times \{1, ..., 1000\}, i < 10^6 \longrightarrow \hat{y}(x)?$$

"Rhinos"

Estimation task

Training set to predict labels

"Rhino"

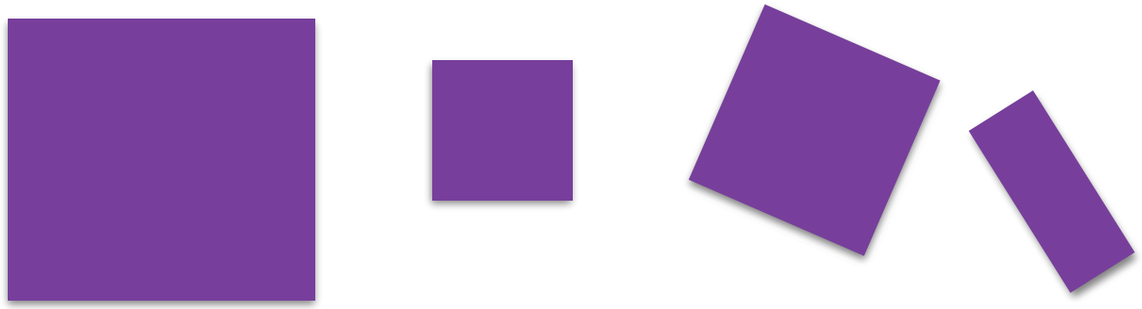
Not a "rhino"

. . . hard problems:

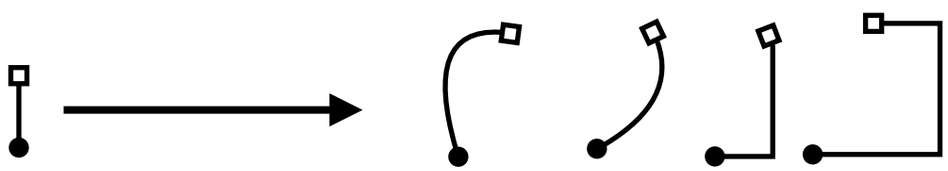
Image variabilities

Geometric variability

Groups acting on images:
translation, rotation, scaling



Other sources : luminosity, occlusion,
small deformations



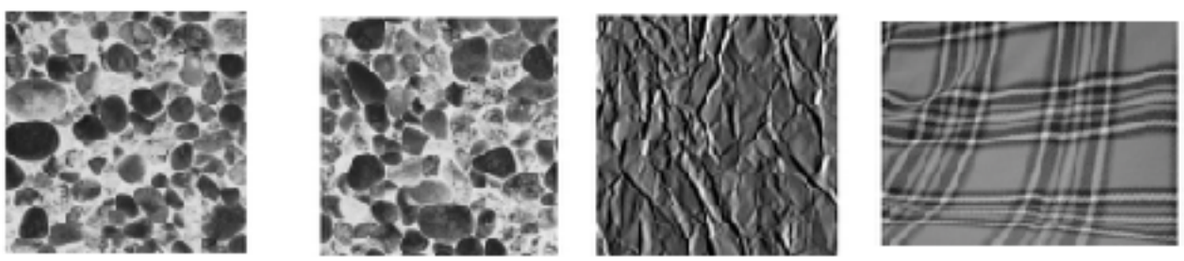
High variance: hard to reduce!

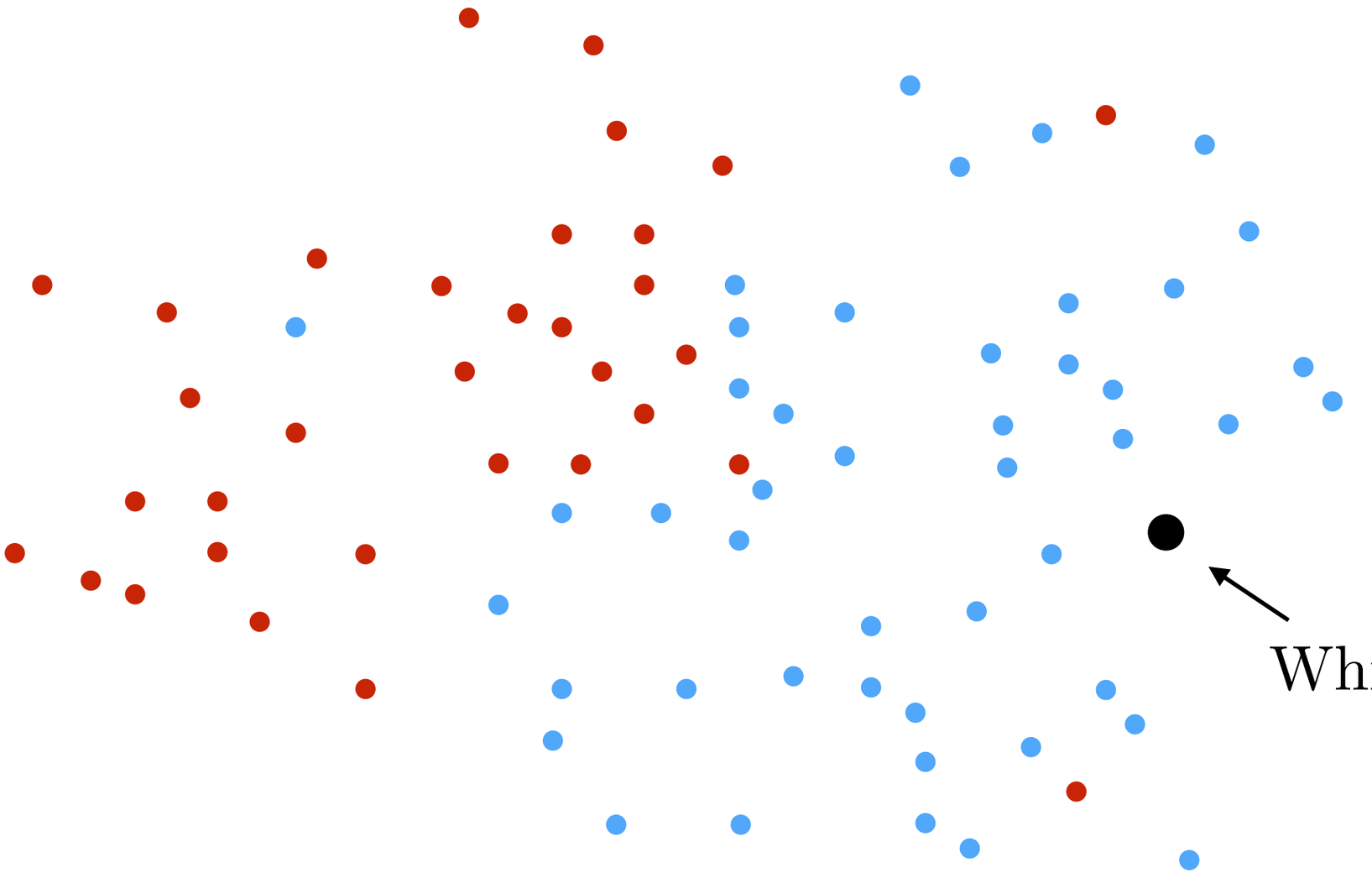
Class variability

Intraclass variability

Not informative

Extraclass variability





Counter-intuitive picture!

An example of supervised problems: classification

Breaking the curse of dimensionality with Deep Learning

- With deep learning, one simply input the data, with limited a priori on the data structures:
  - convolutions
  - data augmentations...
  ... in setting where there is no data model.

- And the tasks are **solved**!

- **Why, how?**

Supervised tasks

- In deep-learning, a key ingredient is the supervision:

For some $(x, f(x))$, one looks for: $\Phi x \approx f(x)$

Very high dimensional

Often mono dimensional

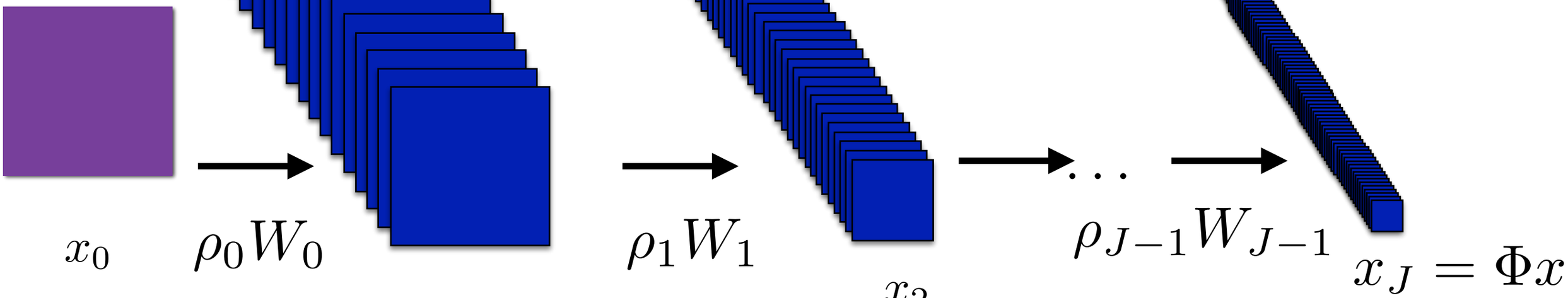
To be estimated

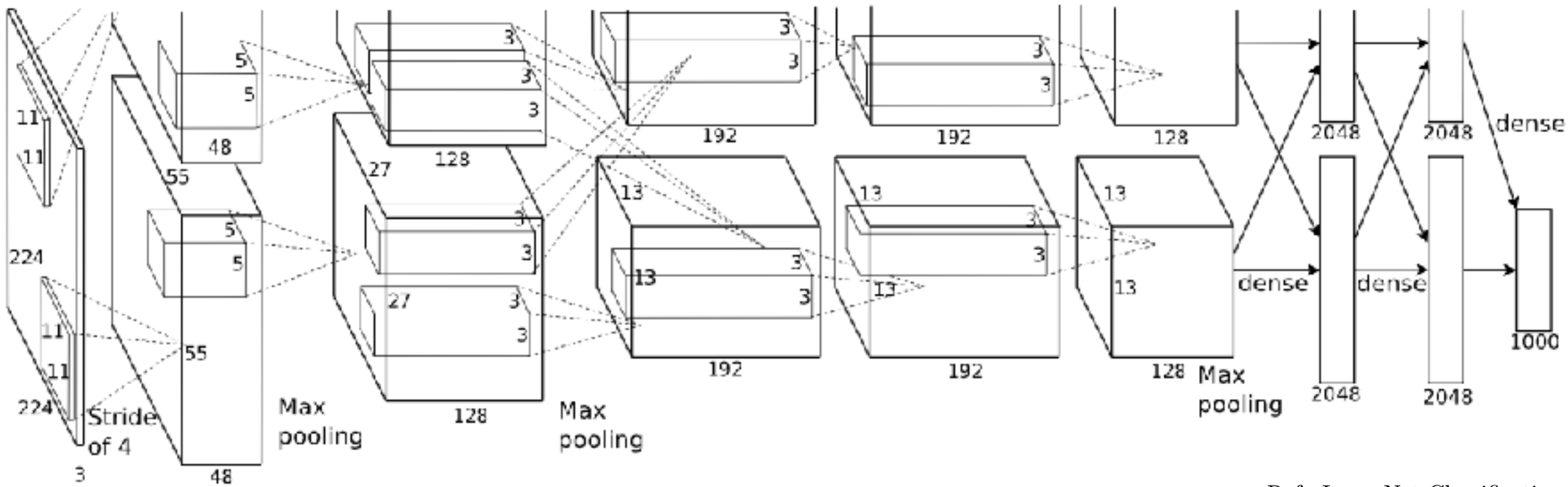
requires to understand the regularity of f

non-linear operator

convolutional operator

$$x_{j+1} = \rho_j W_j x_j$$





Deep Convolutional Neural Network

Ref.: ImageNet Classification with Deep Convolutional Neural Networks. A Krizhevsky et al.

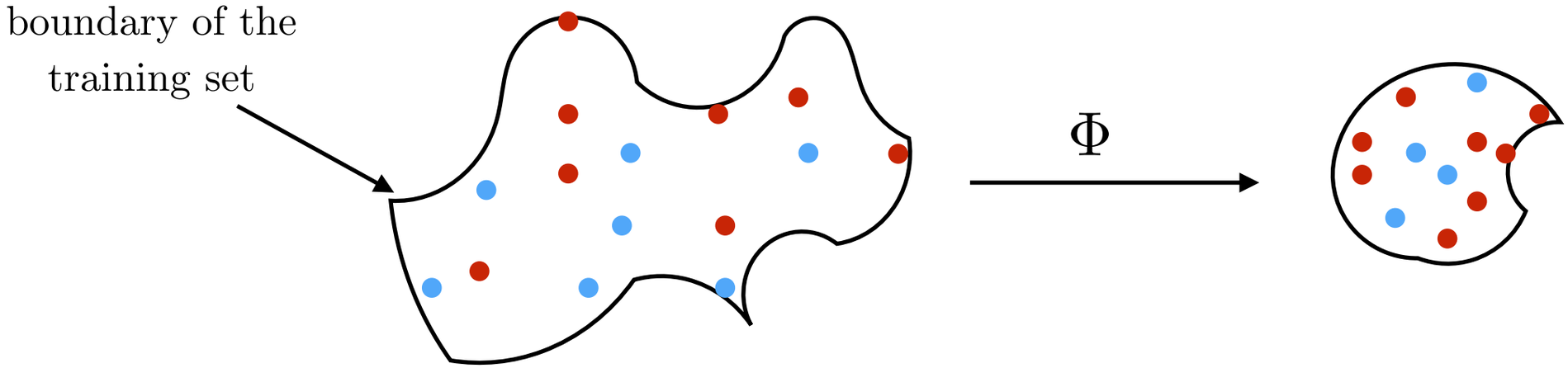
Necessary mechanism:

Separation - Contraction

- In high dimension, typical distances are huge, thus an appropriate representation must contract the space:

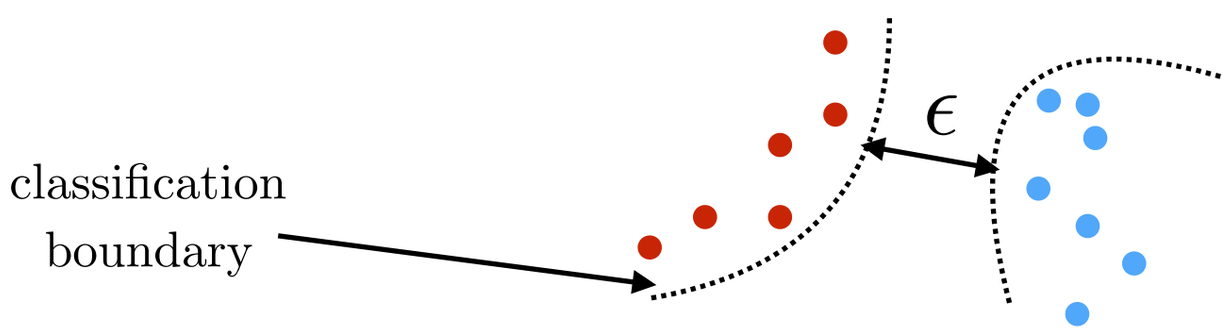
$$\|\Phi x - \Phi x'\| \leq \|x - x'\|$$

Ref.: Understanding deep convolutional networks
S Mallat



- While avoiding the different classes to collapse:

$$\exists \epsilon > 0, y(x) \neq y(x') \Rightarrow \|\Phi x - \Phi x'\| \geq \epsilon$$



Universal approximation?

- Let f be a compactly smooth supported function:

  $f : \mathbb{R}^D \to \mathbb{R}$ and a smoothness measure: $C_f = \int_{\mathbb{R}^D} \|\omega\|_1 \ |\hat{f}(\omega)| \, d\omega$

  and ρ a non-linearity, bounded, strictly monotically increasing and continuous(e.g. tanh)

- Theorem: Universal approximation (Cybenko, 1991)

Let's note: $F^P : \{a_i, w_i\}_{i \leq P}$ and $F^P(x) = \sum_{i \leq P} a_i \rho(w_i^T x + b_i)$

Then: $\forall \epsilon, \exists F^P : \|F^P - f\|_\infty < \epsilon$

- Theorem: Approximation and estimation bounds (Barron, 1994)

If: $F^{N,P} = \arg\inf_{F^P} \sum_{j=1}^{N} \|F^P(X_j) - f(X_j)\|^2$

then: $E\|F^{N,P} - f\|^2 \leq \mathcal{O}(\frac{C_f^2}{N}) + \mathcal{O}(\frac{DN}{P} \log(P))$

Approximation issues

- Yet:
  - Typically, $N = \mathcal{O}(\epsilon^{-D})$
  - No method to get the $(a_i, w_i)_{i \leq N}$
  - 1-layer does not correspond to typical applications

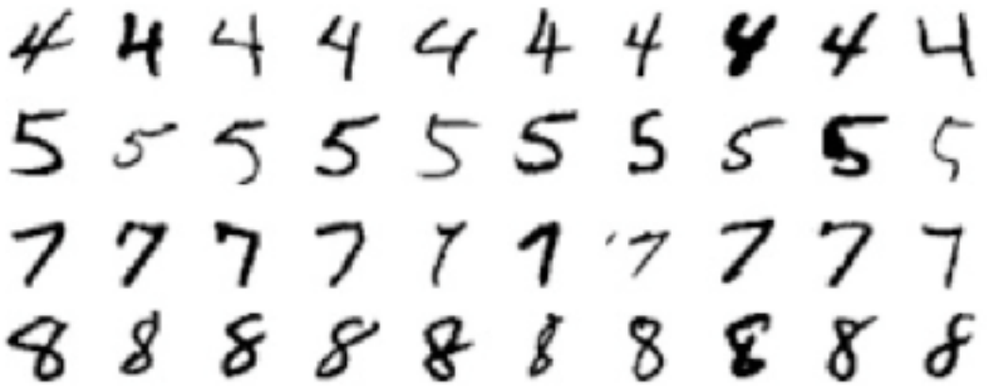
Model?

- Which model applies on the data?

- What are the mechanisms in the inner layers?

Model on the data: low dimensional manifold hypothesis?

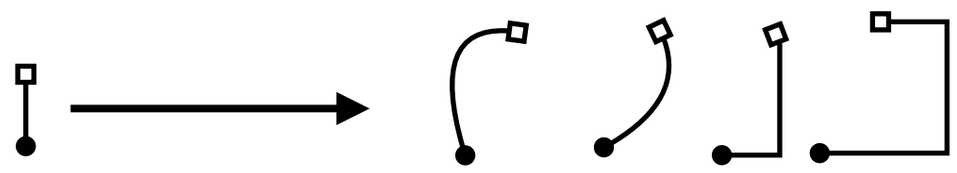
- Low dimensional manifold: dimension up to 6. Not higher:
  Property: if $f : \mathbb{R}^D \to [0,1]$ is 1-Lipschitz, then let
  $N_\epsilon = \arg\inf_N \sup_{i \leq N} \left(|f(x) - f(x_i)| < \epsilon \right)$.
  Then $N_\epsilon = \mathcal{O}(\epsilon^{-D})$
- Can be true for MNIST...



All variabilities are known

Small "limited" deformations +Translation

- High dimensional deformations in the general case!



Flatenning the space: progressive manifold?

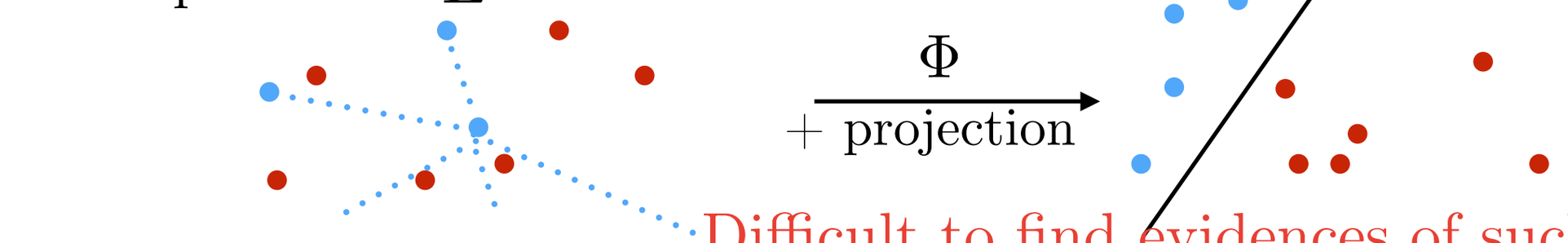
- Parametrize variability on synthetic data: $L_\theta, \theta \in \mathbb{R}^d$ and observe it after PCA

Ref.: Understanding deep features with computer-generated imagery, M Aubry, B Russel

(c) Object color (d) Background color (a) Lighting (b) Scale

- Data tends to live on flattened space. Tangent space?

$$\sup_L \frac{\|\Phi L x - \Phi x\|}{\|Lx - x\|} < \infty \Rightarrow \exists \text{ "weak" } \partial_x \Phi$$
$$\Rightarrow \Phi L x \approx \Phi x + \underbrace{\partial_x \Phi L}_{\text{A linear operator}} + o(\|L\|)$$

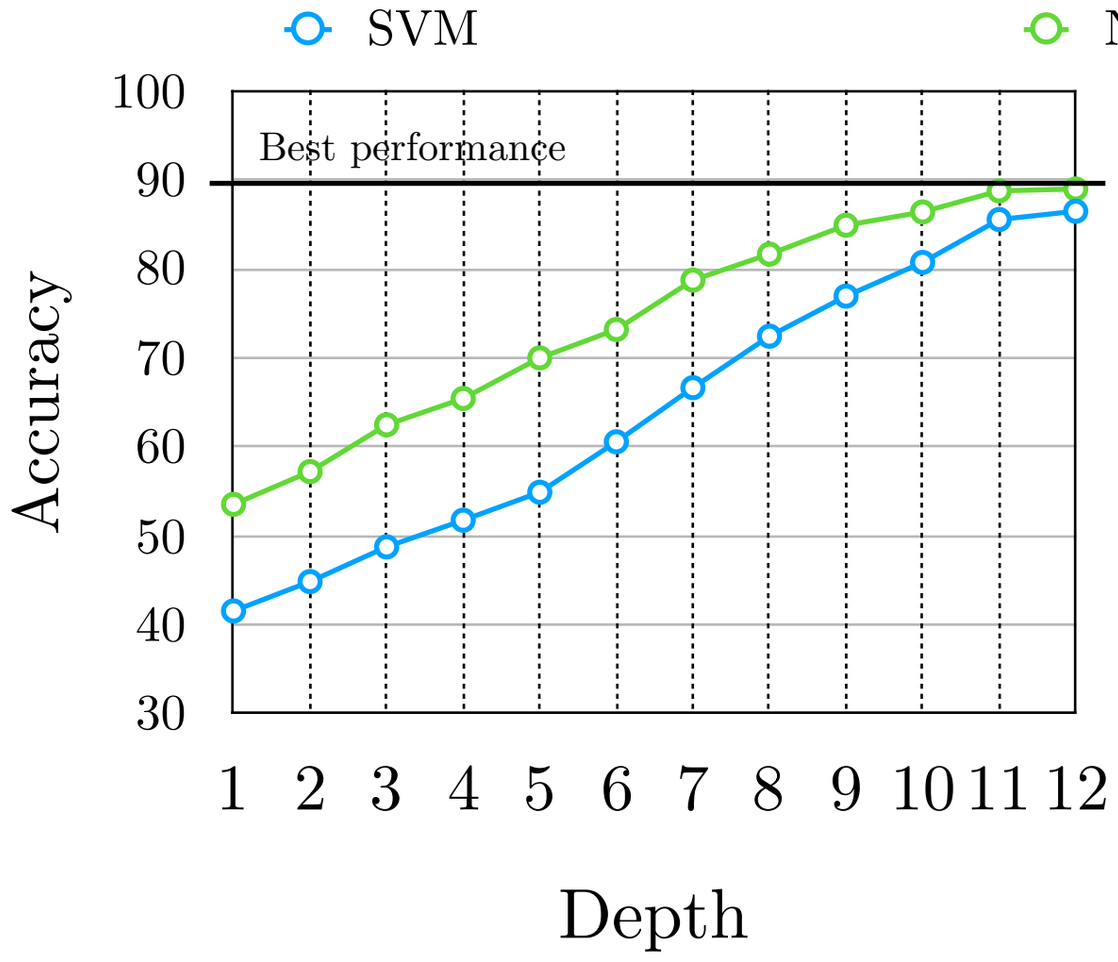


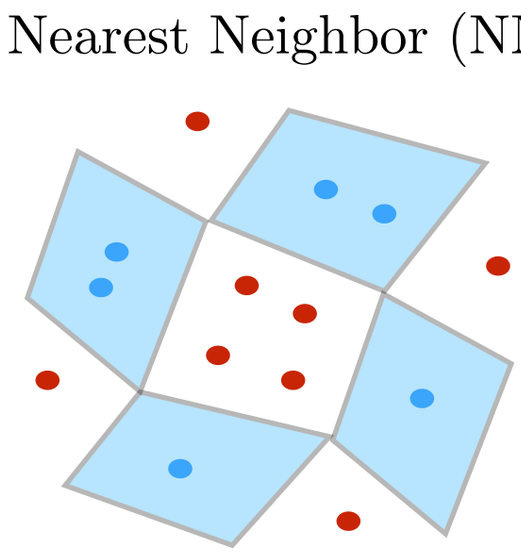
Difficult to find evidences of such phenomenon

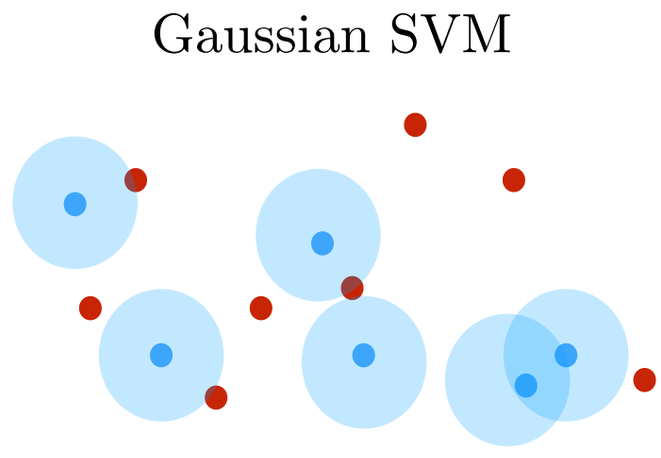
Progressive linearisation and contraction

- We aim to show the cascade permits a progressive contraction & separation, w.r.t. the depth:

In the following, representations are spatially averaged.







Localised classifiers

- How can we explain it?

Ref.: Building a Regular Decision Boundary with Deep Networks, EO

Reducing mutual information (Information bottleneck)

$$I(X;Y) = \int_{\mathbb{R}^2} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \mathrm{d}x\mathrm{d}y = H(X) - H(X|Y)$$

Ref.: Opening the Black Box of Deep Neural Networks via Information, R. Shwartz-Ziv and N. Tishby

Measures the dependancy between variables

- Reducing the information sounds relevant:

$$I(X;\Phi_1 X) \geq I(X;\Phi_2 X) \geq ... \geq I(X;\Phi_J X)$$

"Compress" X

$$I(X;Y) \geq I(\Phi_1 X;Y) \geq ... \geq I(\Phi_J X;Y)$$

... but "reveal" Y

They propose to introduce:

$$\Phi_{j,\lambda} = \arg\inf_{\Phi} I(\Phi_{j-1}X, \Phi_j X) - \lambda I(\Phi_j X, Y)$$

- But one can easily build invertible CNNs...

Generalization?

- How can we explain CNN good generalization?
- The amount of data is small w.r.t. the dimensionality!!

$$N = \mathcal{O}(\epsilon^{-D})$$

Complexity results, Vapnik tools

Empirical Rademacher complexity:

$$\mathcal{R}_{\mathcal{F}}(z_1, ..., z_n) = E_\sigma[\sup_{f \in \mathcal{F}} \frac{1}{N} \sum_{i=1}^{N} \sigma_i f(z_i)] \qquad \sigma_i \in \{-1, 1\}$$

- Typical decomposition of generalisation:

$$E_z(\text{error(z)}) - \widehat{\text{error}(z)} \leq \mathcal{R}(z) + \mathcal{O}(\sqrt{\frac{\ln(\frac{1}{\delta})}{N}})$$

$$\text{where } 0 \leq \text{error}(z) \leq 1$$

- In fact, it is empirically shown that CNN can fits random labels... Thus:

Ref.: Understanding Deep Learning requires rethinking generalization, C Zhang et al.

$$\mathcal{R}_{\mathcal{F}}(z) \approx 1$$

Margins (barlett)

Define a margin: (error)

Ref.:Spectrally-normalized margin bounds for neural networks, P Barlett et al.

$$\mathcal{M}(x,y) \triangleq (L\Phi x)_y - \max_{i \neq y} (L\Phi x)_i$$

- Theorem (Bartlett,): With high probability, if $\mathcal{R}(\Phi) \leq r$

$$\mathbb{P}(\mathcal{M}(x,y) \leq 0) \leq \frac{1}{N} \sum_{n=1}^{N} 1_{\mathcal{M}(x_n,y_n) \leq \gamma} + O(\frac{r}{\gamma\sqrt{N}})$$

Requires margin existence...

where:
$$\mathcal{R}(\Phi) = \prod_j \|W_j\|_* \sqrt{\sum_j \frac{\|W_j\|_F}{\|W_j\|_*}}$$

Scale sensitive + but no linear dependancy in #params

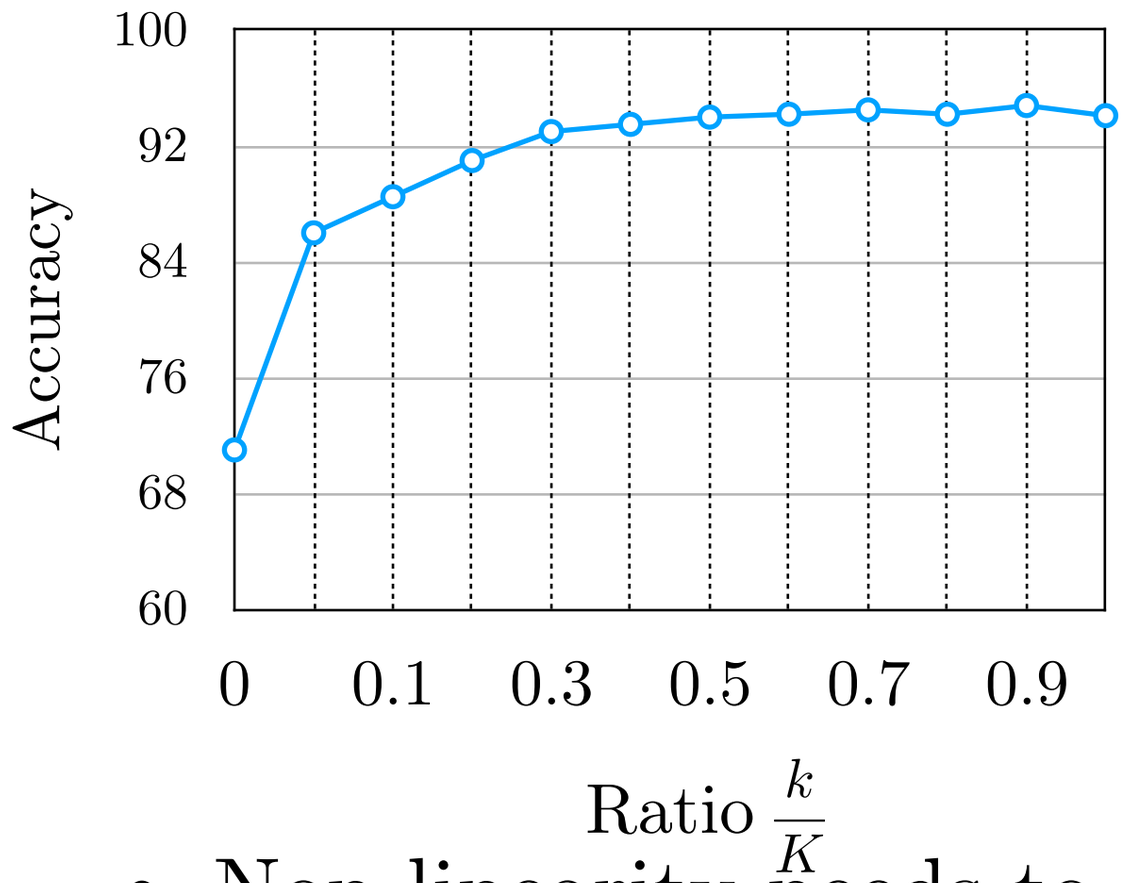
Architectures?

- What is the role of the non-linearity?

- What should be chosen, depth or width?

The non-linearity: to contract?

- "More non-linear is better."



ReLU on a fraction $\frac{k}{K}$ of the coefficients a layer x:

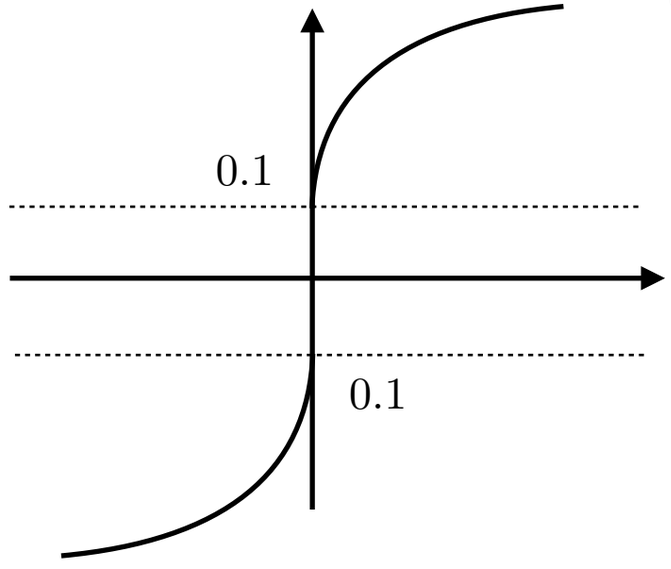
$$\text{ReLU}_k^K(x)(.,l) \triangleq \begin{cases} \text{ReLU}(x(.,l)), & \text{if } l \leq k \\ x(.,l), & \text{otherwise} \end{cases}$$

Tradional pointwise non-linearity can be weakened

Ref.: Building a Regular Decision Boundary with Deep Networks, EO

- Non-linearity needs to contract or being continuous?

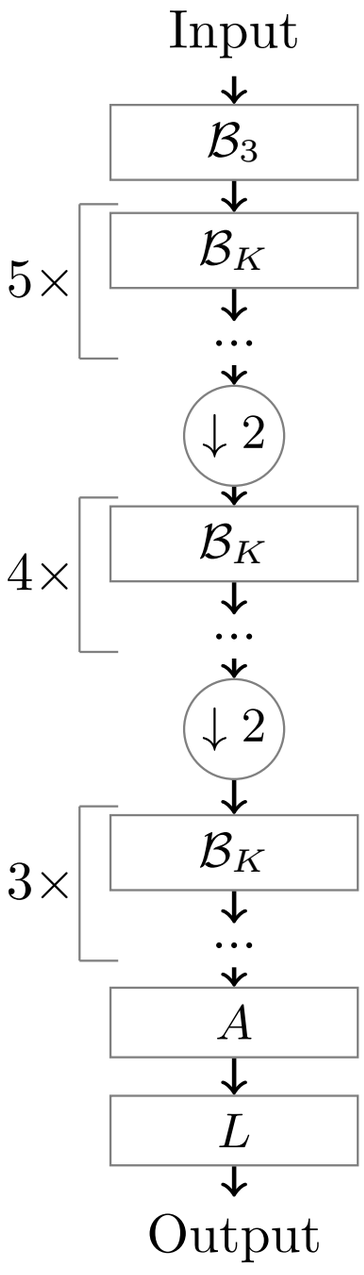
$$\rho(x) = \text{sign}(x)(\sqrt{|x|} + 0.1)$$

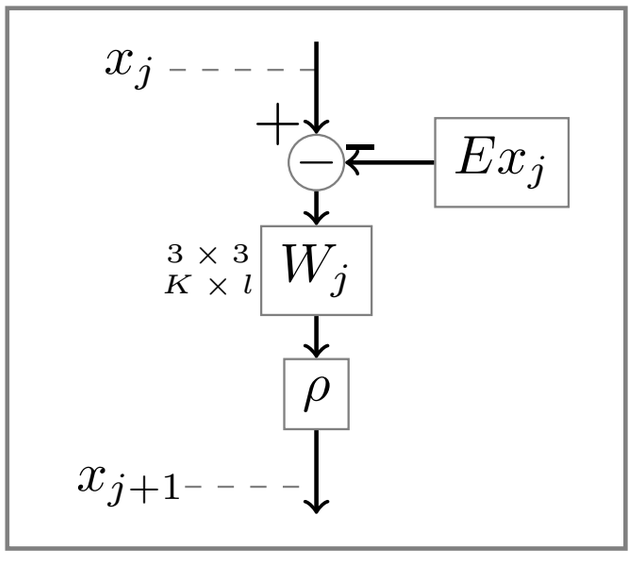


good perf (89% acc.) on CIFAR10

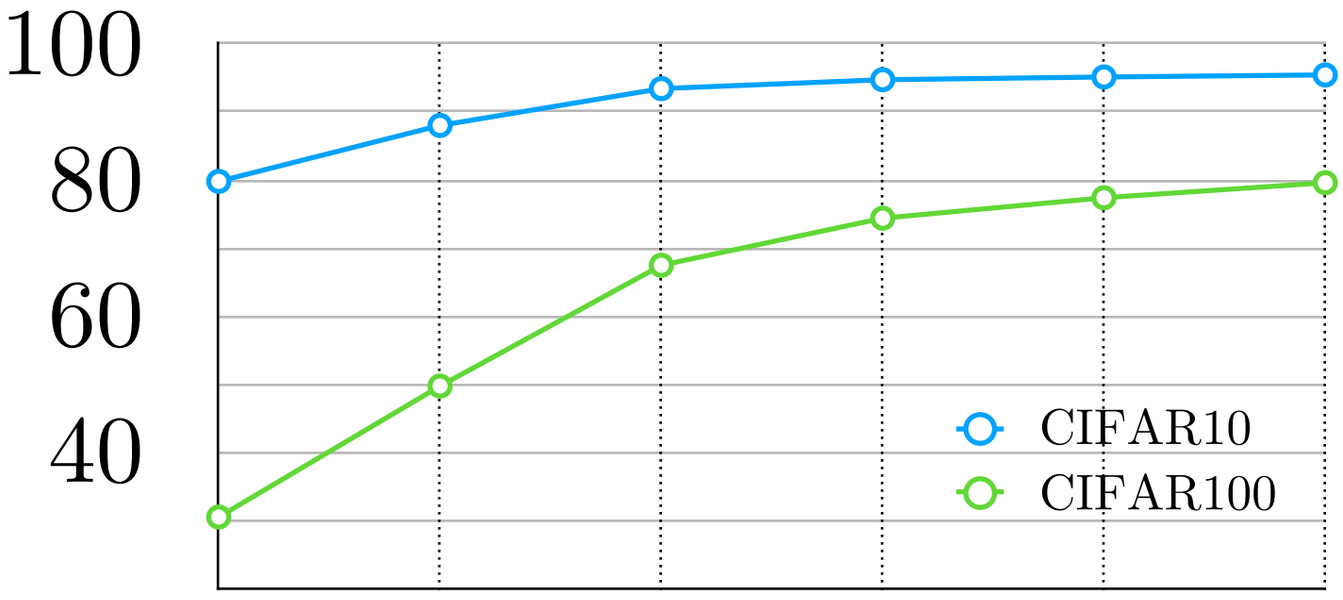
Depth/width?

- Current architecture tends to maintain the input size: is there a reason?





A block $\mathcal{B}_l$



$$x_{j+1} = \rho W_j(x_j - Ex_j)$$

BatchNorm

Ref.: Building a Regular Decision Boundary with Deep Networks, EO

- Width/depth: WideResNet-50 vs ResNet-200

Ref.: Wide Residual Networks, S Zagoruyko and N Komodakis

A result on depth

Ref.: The Power of Depth for Feedforward Neural Networks, R Eldan and O Shamir

- Theorem: there exists g compact bounded continuous *such that:*

- A 3-layer Neural Network can aproximate g with polynomial number of neurons in the dimension

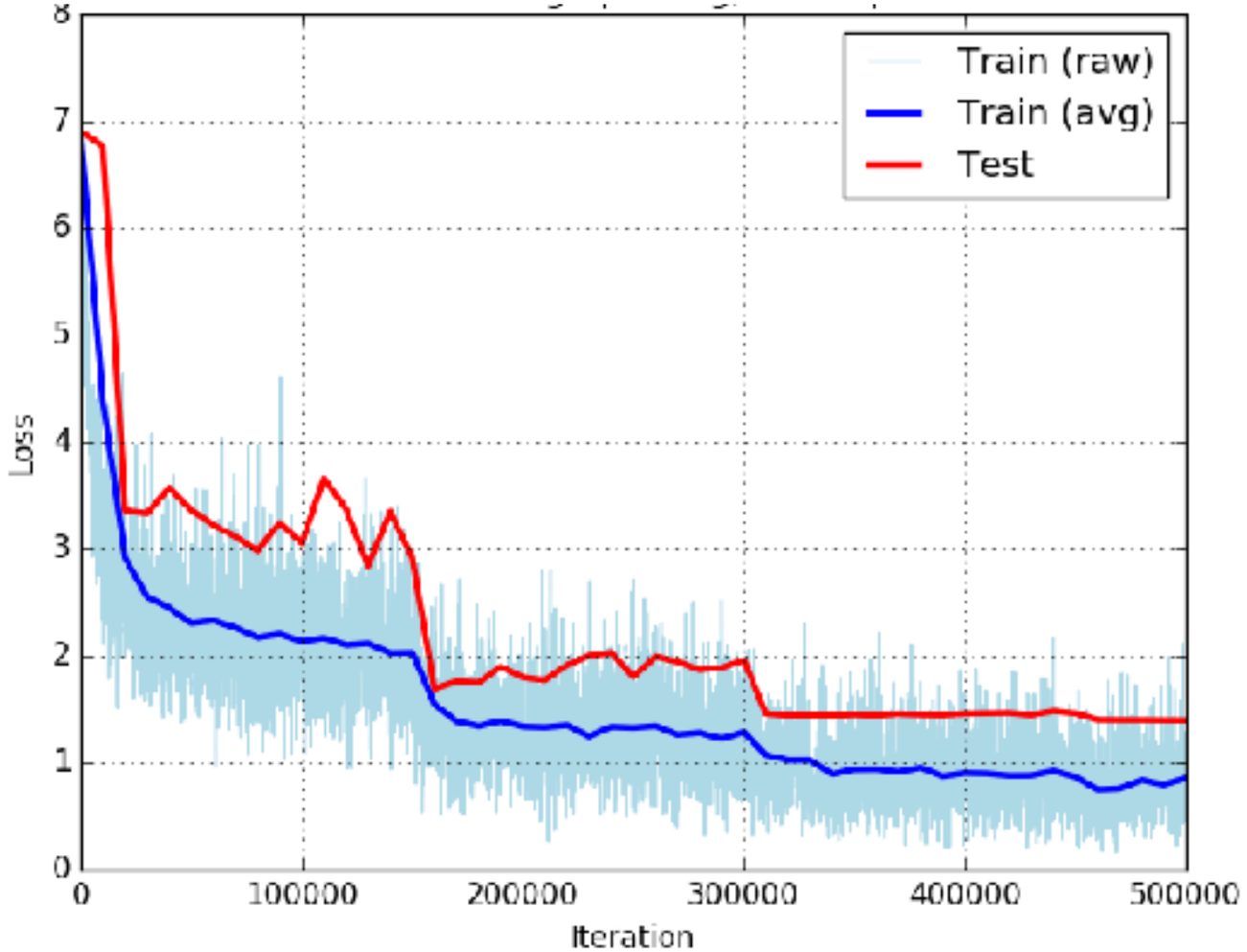
- No 2-layer Neural Network can approximate g with less than an exponential number of neurons in the dimension

Optimisation of a CNN

- Contrary to the convex case with appropriate models, here there is:
  - no guarantee of convergence
  - no guarantee of generalisation
  - only a limited understanding of regularization

Optimizing a CNN

- How can we explain the learning rate schedule?
- Why do we need to train during long epoch?
- Conditioning techniques, weight decay, BatchNorm... ?



Example of symmetry

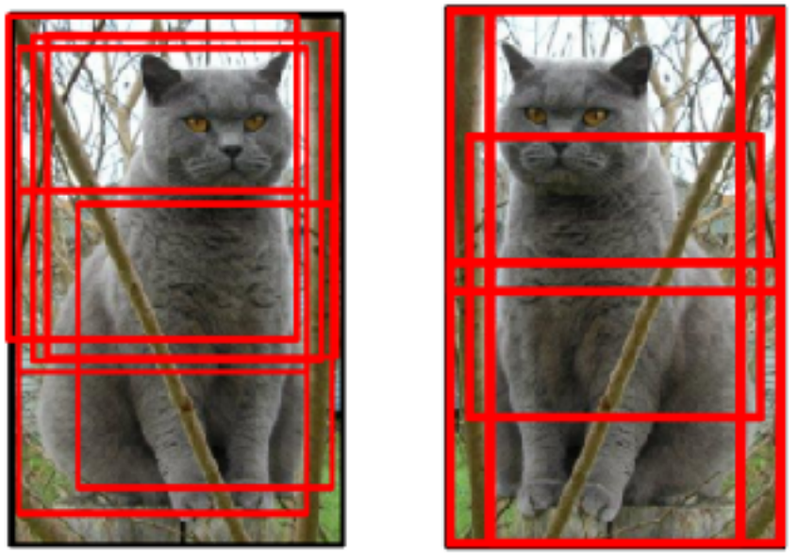
Data augmentation

- A good way to introduce symmetry: data augmentation (is it really creating new data?)

$$\inf_{\Phi} \mathbb{E}[\text{loss}(\mathcal{L}, X, Y, \Phi)]$$

with for ex.:

$$\text{loss}(\mathcal{L}, X, Y, \Phi) = \|\Phi\mathcal{L}X - Y\|^2$$



Flip+random translation

- In this case, ideally:

$$\Phi\mathcal{L}x \approx \Phi x$$

What does the CNN see?

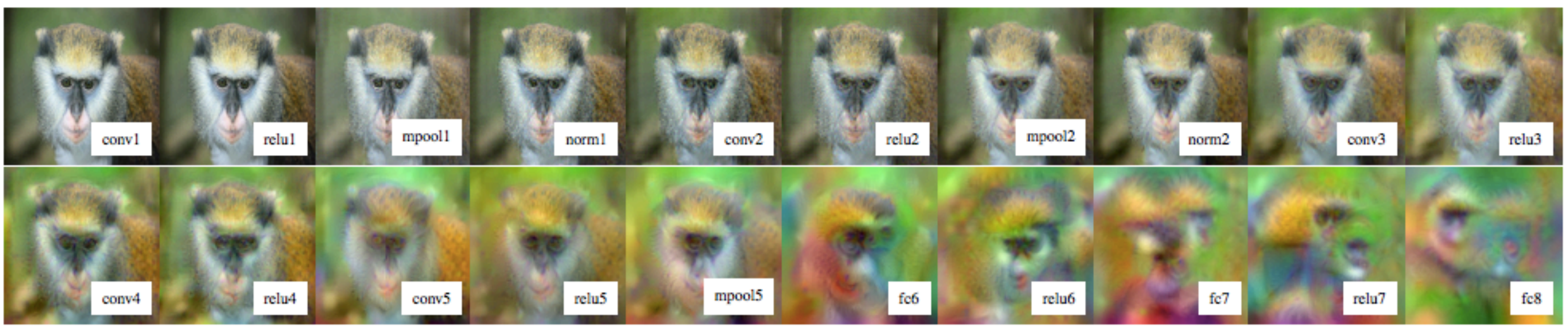
- How does the memory of a CNN work?
- Template deformable matching model?
- Sample memorization?

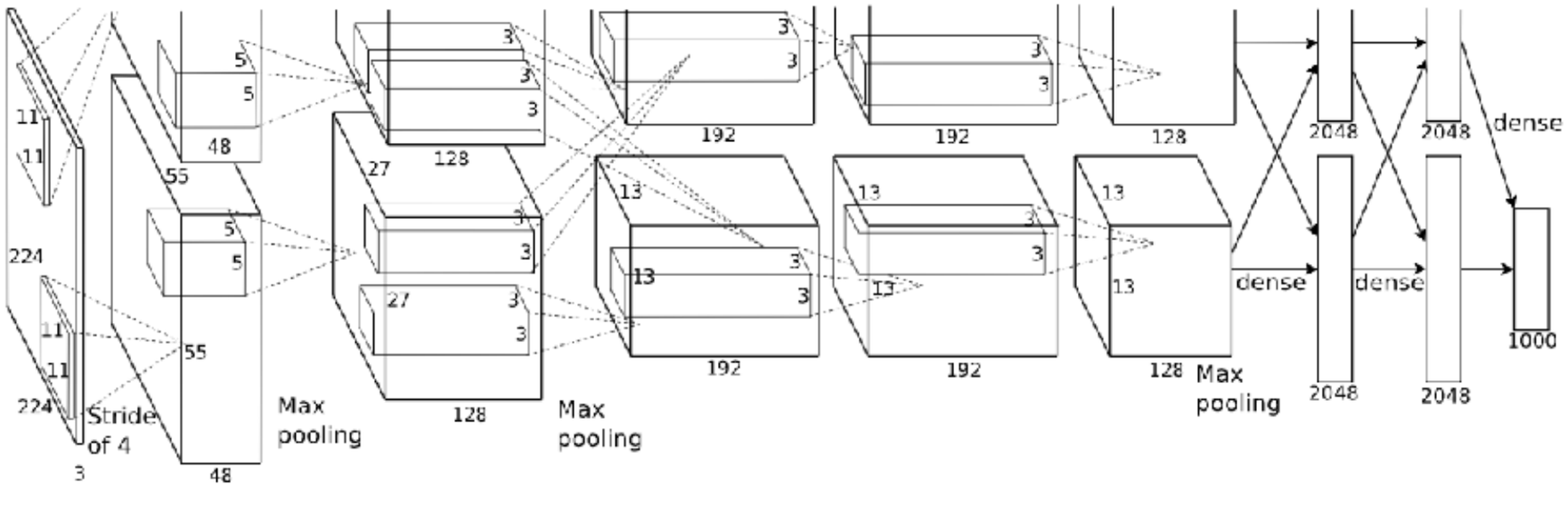
Visualizing(1): the representations learned

- Reconstructing:

$$\inf \|\Phi_j x - \Phi_j y\| + \mathcal{R}(y)$$

Ref.: Visualizing Deep Convolutional Neural Networks Using Natural Pre-Images, A Mahendran and A Vedaldi





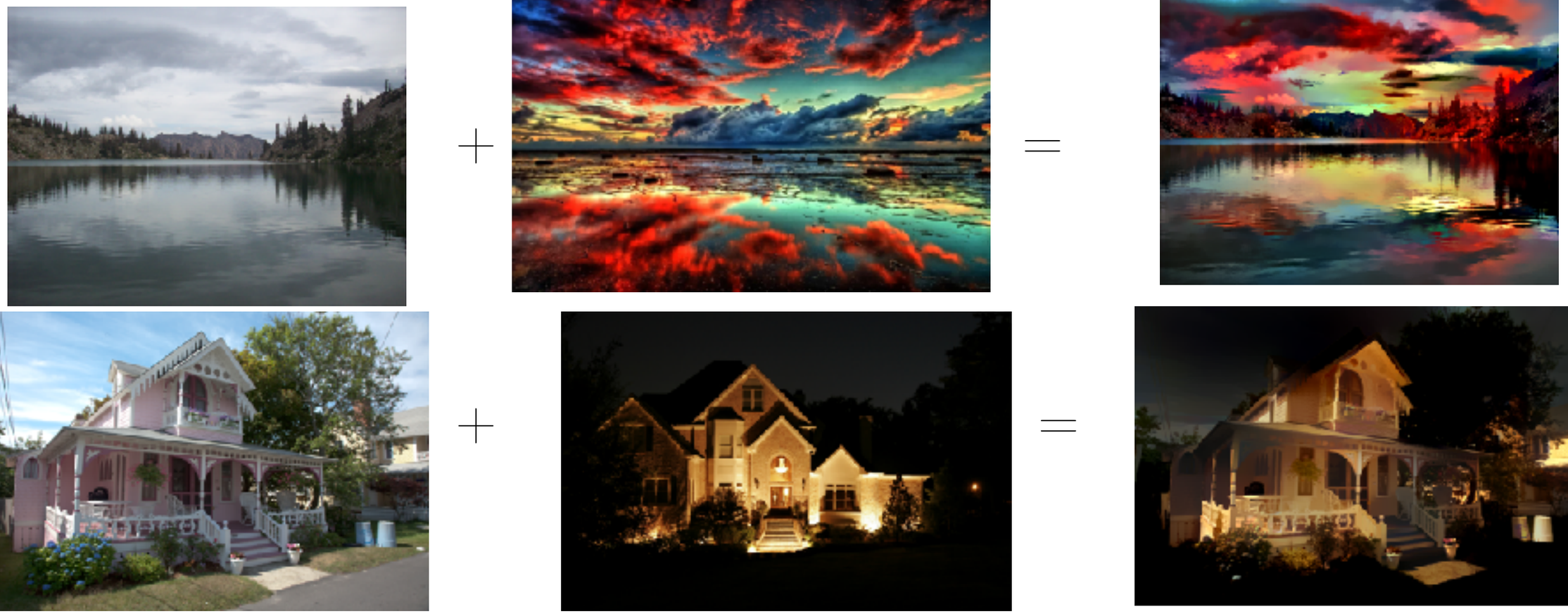
Visualizing(2): style transfer

Ref.: Deep Photo Style Transfer, Luan et al.

$$\arg\min_{\tilde{x}} \|\Phi x - \Phi\tilde{x}\|^2 + \lambda\|\mathrm{Cov}(\Phi y)) - \mathrm{Cov}(\Phi\tilde{x})\|^2$$

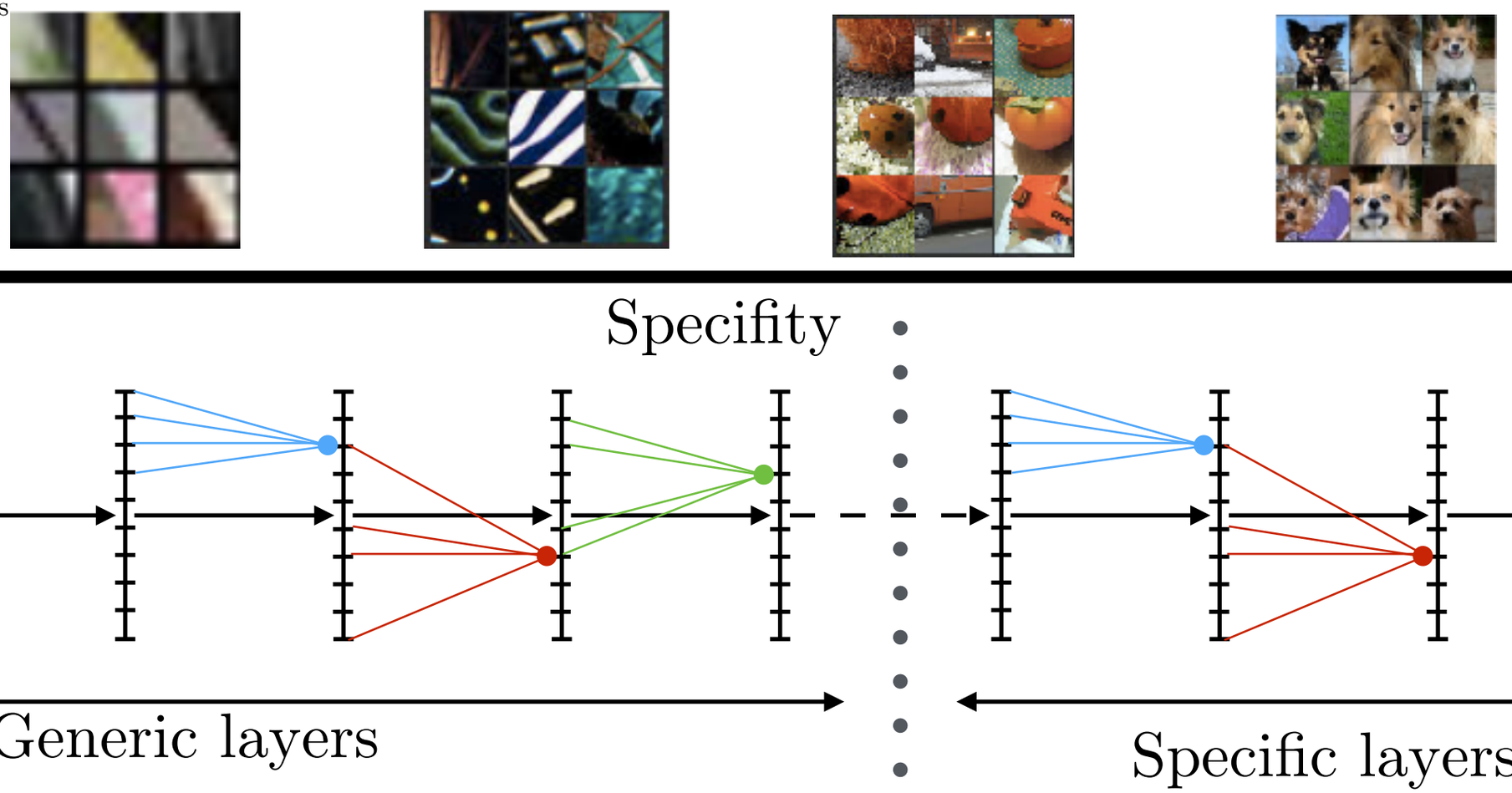
Original image — Style transfer

Input Φx | Target style Φy | Output $\Phi\tilde{x}$



Transferrable representations

Ref.: Visualizing and understanding Convolutional Networks
Zeiler, et al.



- It is possible to retrain the last layers of a Neural Network, freezing the previous one to obtain robust system for classification, *on data that were not observed.*

Ref.: How transferable are features in deep neural networks?
Yozinski et al.

- Genericity & geometry in the early layer?

Incorporating structures to analyse

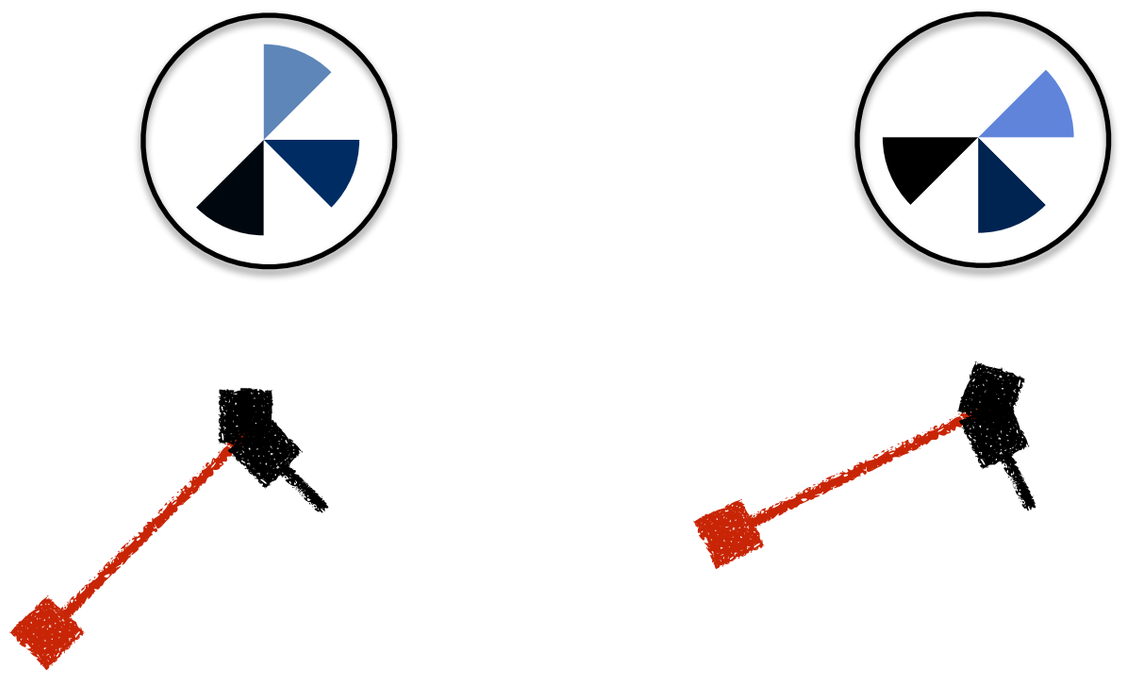
- Let us consider:

$$x \in \ell^2, Sx(\theta) \text{ such that: } Sr\_{\tilde{\theta}}x(\theta) = Sx(\theta + \tilde{\theta})$$

- and discretize it:

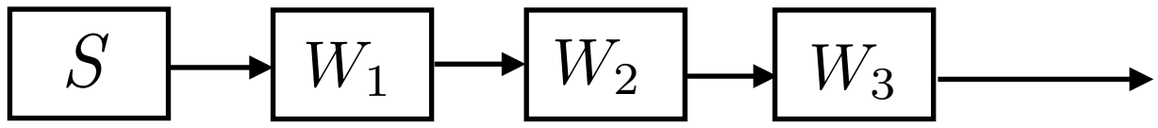
$$\theta \in \{\frac{2k\pi}{n}\}_{k \leq n}$$

- S is covariant with rotation:



Invariance learned, rotation

- Consider a CNN learns on top of S:



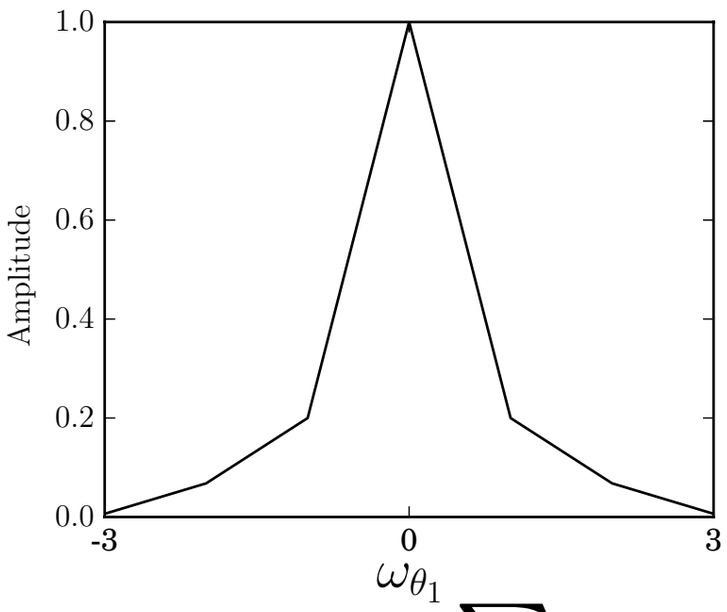
- The operator W1 is structured as follow:

$$(W_1 S)_k = \sum_{\theta \in \Theta} w_{k,\theta}$$

Ref.: Scaling the Scattering Transform:
Deep Hybrid Networks
EO, E Belilovsky, S Zagoruyko

One can thus analyse the Fourier transform of w:

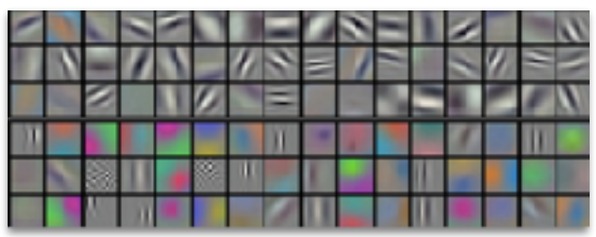
$$\hat{f}(\omega_\theta) = \int_{[0,2\pi]} f(\theta) e^{-i\omega_\theta \theta} \, d\theta$$



$$\Omega(W_1)(\omega_\theta) = \sum_k |\hat{w}_{k,\omega_\theta}|^2$$

It's a low pass!
Invariant to rotation.

method: similar to AlexNet
first layer analysis



The concept of neurons, a good concept?

Ref.: Intriguing properties of neural networks
C Szegedy et al.

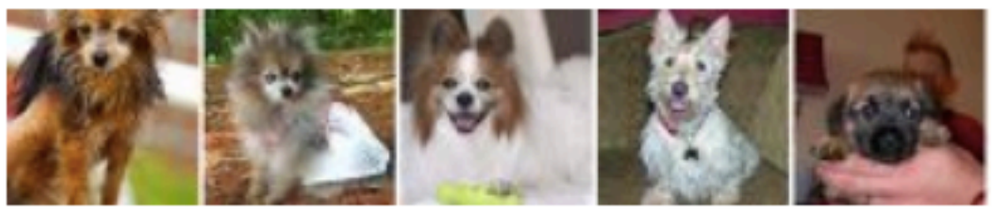
- Intriguing properties of CNNs

$$x' = \arg \max_{x \in \{\text{training set}\}} \langle \Phi x, w \rangle \quad \text{for any } w$$

(a) Direction sensitive to white, spread flowers.

(b) Direction sensitive to white dogs.

(c) Direction sensitive to spread shapes.



(d) Direction sensitive to dogs with brown heads.

Challenges in deep learning

- Limited data?
- Formalizing CNNs?
- Interpreting CNNs?
- Do we need to learn every layer?

Conclusion

- Trying to open the blackbox could help to engineer better CNNs
- And give new insights on high dimensional tasks!
- Email me for internships... edouard.oyallon@centralesupelec.fr