What are we learning: visualising the filters and response, memory, wavelets, and attention sinks

Mathematical Institute

Theories of Deep Learning: C6.5,
Lecture / Video 11

Prof. Jared Tanner
Mathematical Institute
University of Oxford

Oxford Mathematics

LeNET-5, an early Image processing DNN:

Network architectures often include fully connected and convolutional layers

UNIVERSITY OF OXFORD
Mathematical Institute

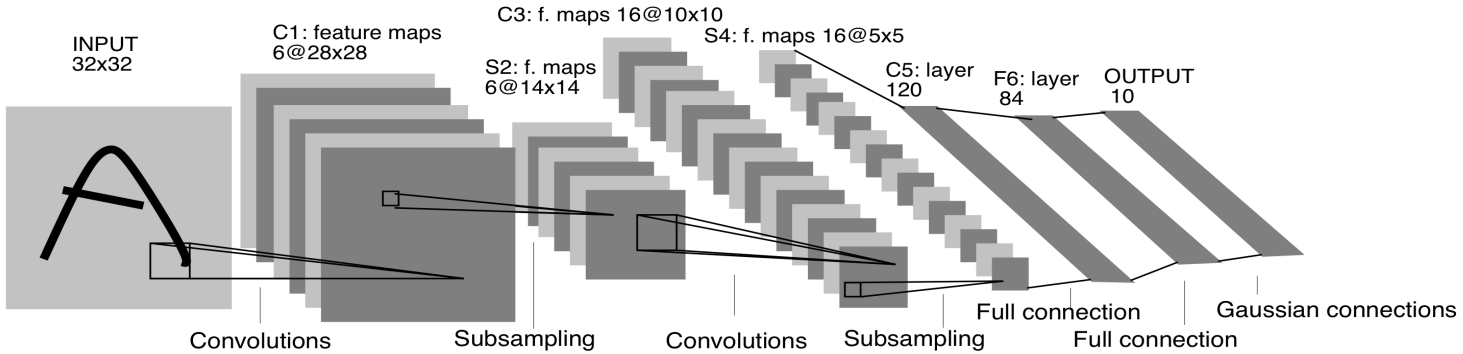


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

C1: conv. layer with 6 feature maps, 5 by 5 support, stride 1.

S2 (and S4): non-overlapping 2 by 2 blocks which equally sum values, mult by weight and add bias.

C3: conv. layer with 16 features, 5 by 5 support, partial connected.

C5: 120 features, 5 by 5 support, no stride; i.e. fully connected.

F6: fully connected, $W \in \mathbb{R}^{84\times 120}$.

`http://yann.lecun.com/exdb/publis/pdf/lecun-98.pdf`

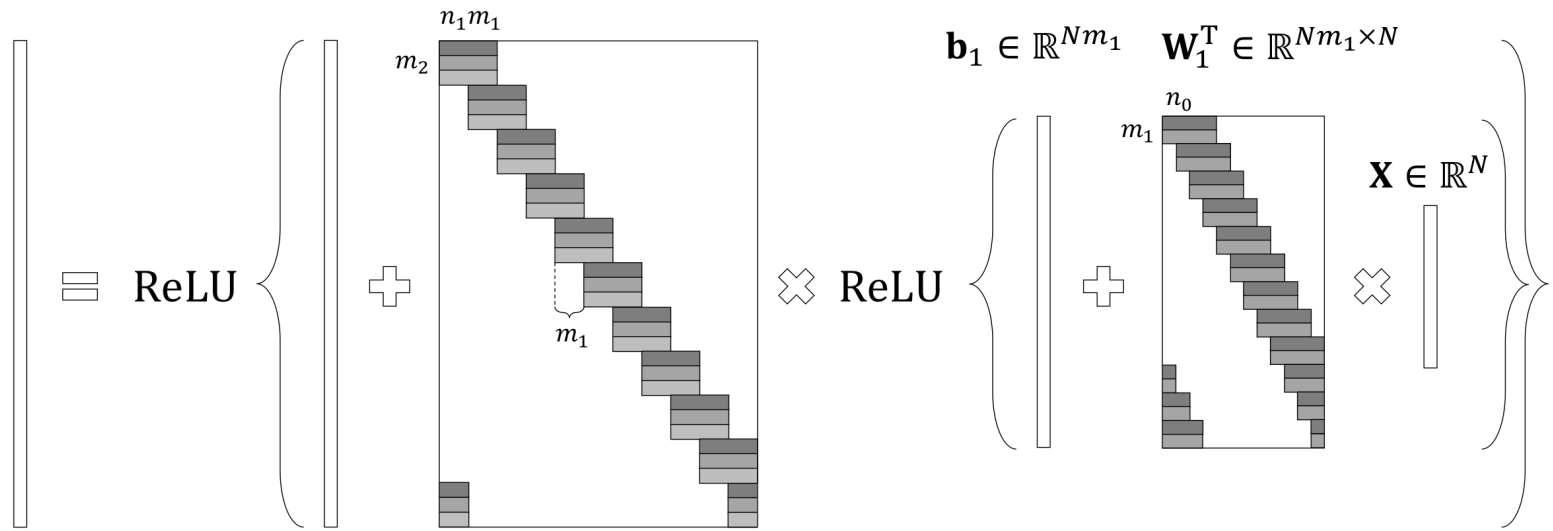
A simple two layer CNN (Papyan et al. 16')

Convolutional structure are the form of multi-resolution analysis

Mathematical Institute

Consider a deep conv. net composed of two convolutional layers:

$\mathbf{Z}_2 \in \mathbb{R}^{Nm_2}$ $\mathbf{b}_2 \in \mathbb{R}^{Nm_2}$ $\mathbf{W}_2^{\mathrm{T}} \in \mathbb{R}^{Nm_2 \times Nm_1}$ $\mathbf{b}_1 \in \mathbb{R}^{Nm_1}$ $\mathbf{W}_1^{\mathrm{T}} \in \mathbb{R}^{Nm_1 \times N}$ $\mathbf{X} \in \mathbb{R}^{N}$



$$Z_2 = \sigma\left(b^{(2)} + (W^{(2)})^T \sigma\left(b^{(1)} + (W^{(1)})^T x\right)\right)$$

https://arxiv.org/pdf/1607.08194.pdf

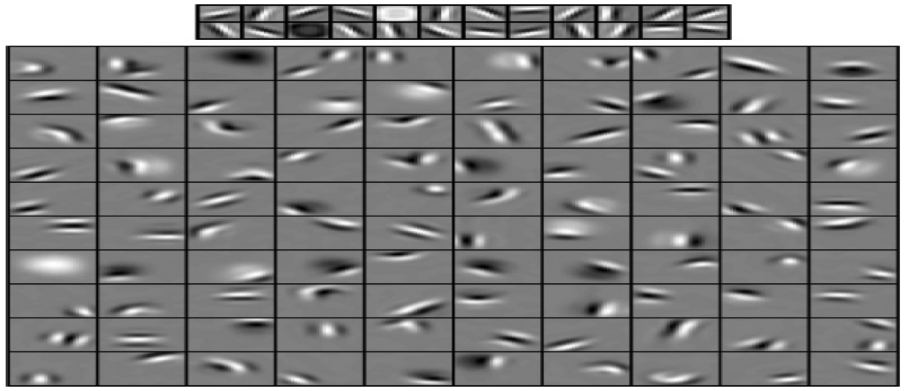
Convolutional Deep Belief Networks (H. Lee et al. 11')

Localized Fourier, Wavelet, structure learned

UNIVERSITY OF OXFORD
Mathematical Institute

We omit the details of this somewhat different architecture, which is stylistically similar to a deep CNN.

Figure 3. The first layer bases (top) and the second layer bases (bottom) learned from natural images. Each second layer basis (filter) was visualized as a weighted linear combination of the first layer bases.



https://cir.nii.ac.jp/crid/1361981469179788288
Display of the convolutional masks in layers 1 and 2, trained from Kyoto natural image database.
http://eizaburo-doi.github.io/kyoto_natim/

Activation maximization (Erhan et al. 09')

Neuron maximization over input space

UNIVERSITY OF OXFORD
Mathematical Institute

For CNNs we can visualize the initial filters. For deeper intermediate layers consider inputs they are most responsive to:

$$x_{i,\ell} = \text{argmax}_x h_i^{(\ell)}(x)$$

with the interpretation that certain neurons have learned to respond to features from the training data.

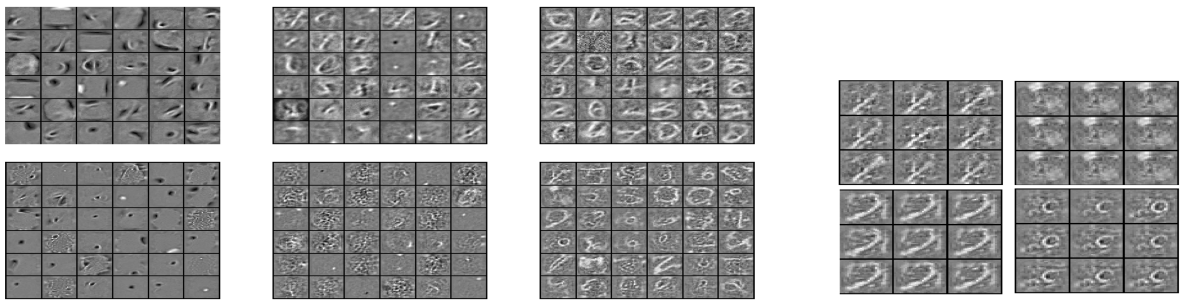


Figure 1: Activation maximization applied on MNIST. On the left side: visualization of 36 units from the first (1st column), second (2nd column) and third (3rd column) hidden layers of a DBN (top) and SDAE (bottom), using the technique of maximizing the activation of the hidden unit. On the right side: 4 examples of the solutions to the optimization problem for units in the 3rd layer of the SDAE, from 9 random initializations.

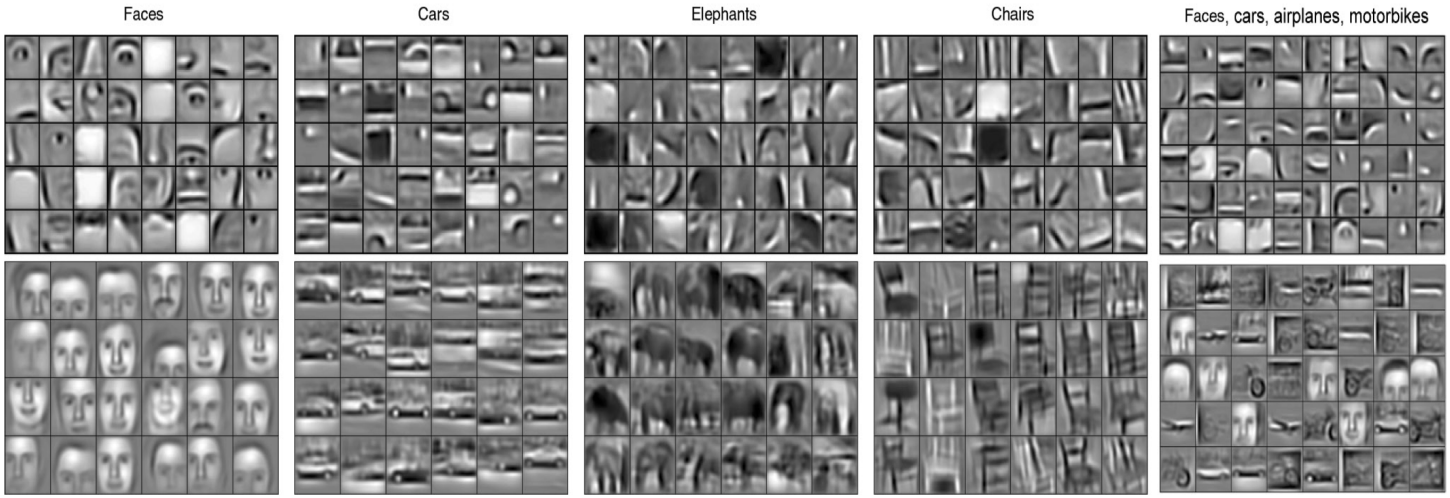
`https://papers.baulab.info/papers/also/Erhan-2009.pdf`

Convolutional Deep Belief Networks (H. Lee et al. 11')

Learned / memorized complex structure from data classes

UNIVERSITY OF OXFORD
Mathematical Institute

Figure 4. Columns 1–4: the second layer bases (top) and the third layer bases (bottom) learned from specific object categories. Column 5: the second layer bases (top) and the third layer bases (bottom) learned from a mixture of four object categories (faces, cars, airplanes, motorbikes).



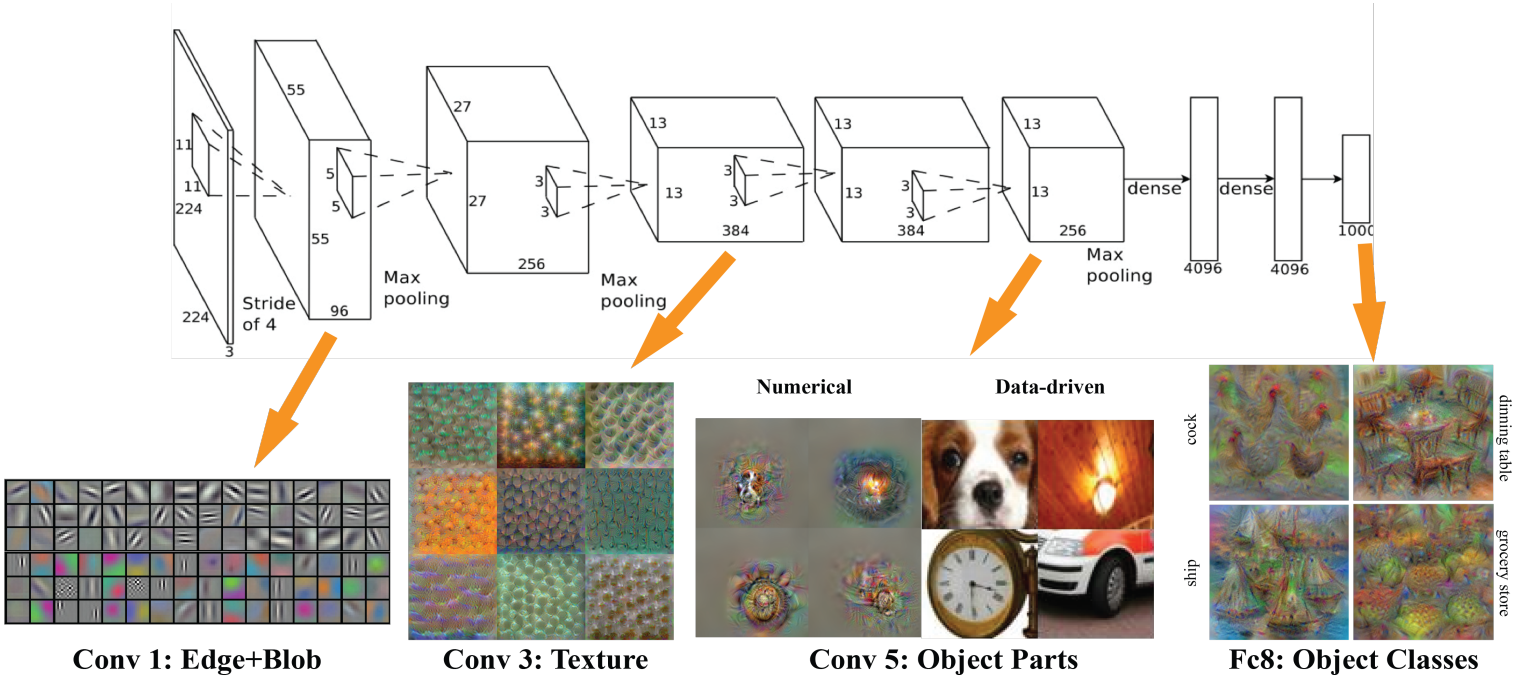
`http://eizaburo-doi.github.io/kyoto_natim/`

The third and fourth layers develop bases which represent features or objects, trained on CalTech 101 dataset.

`http://www.vision.caltech.edu/Image_Datasets/Caltech101/`

Deep CNN, AlexNet (Krizhevsky et al. 12')

Learned / memorized complex structure from data classes



Images are those that maximize specific activation responses.

Layer 1 are masks, subsequent layers are their linear combinations.

`http:`

`//papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`

Deep CNN, VGG (Mahendran et al. 16')

Learned / memorized complex structure from data classes

Mathematical Institute

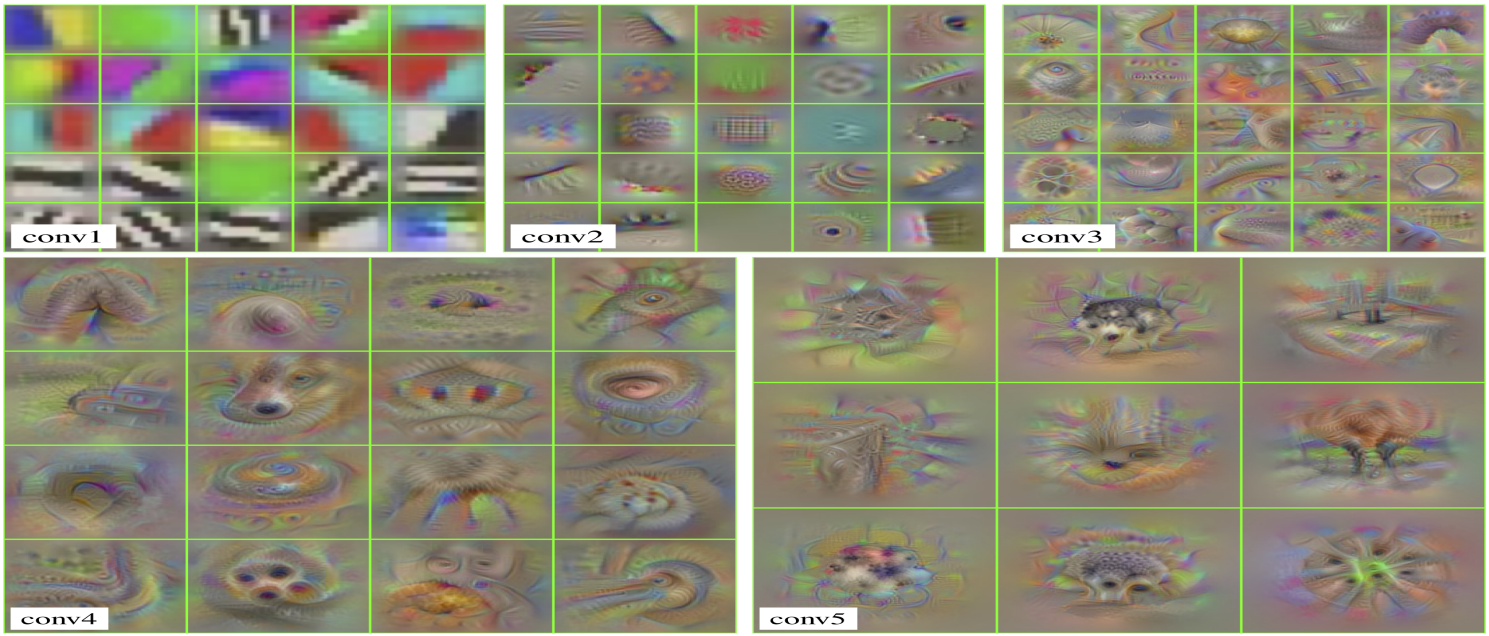


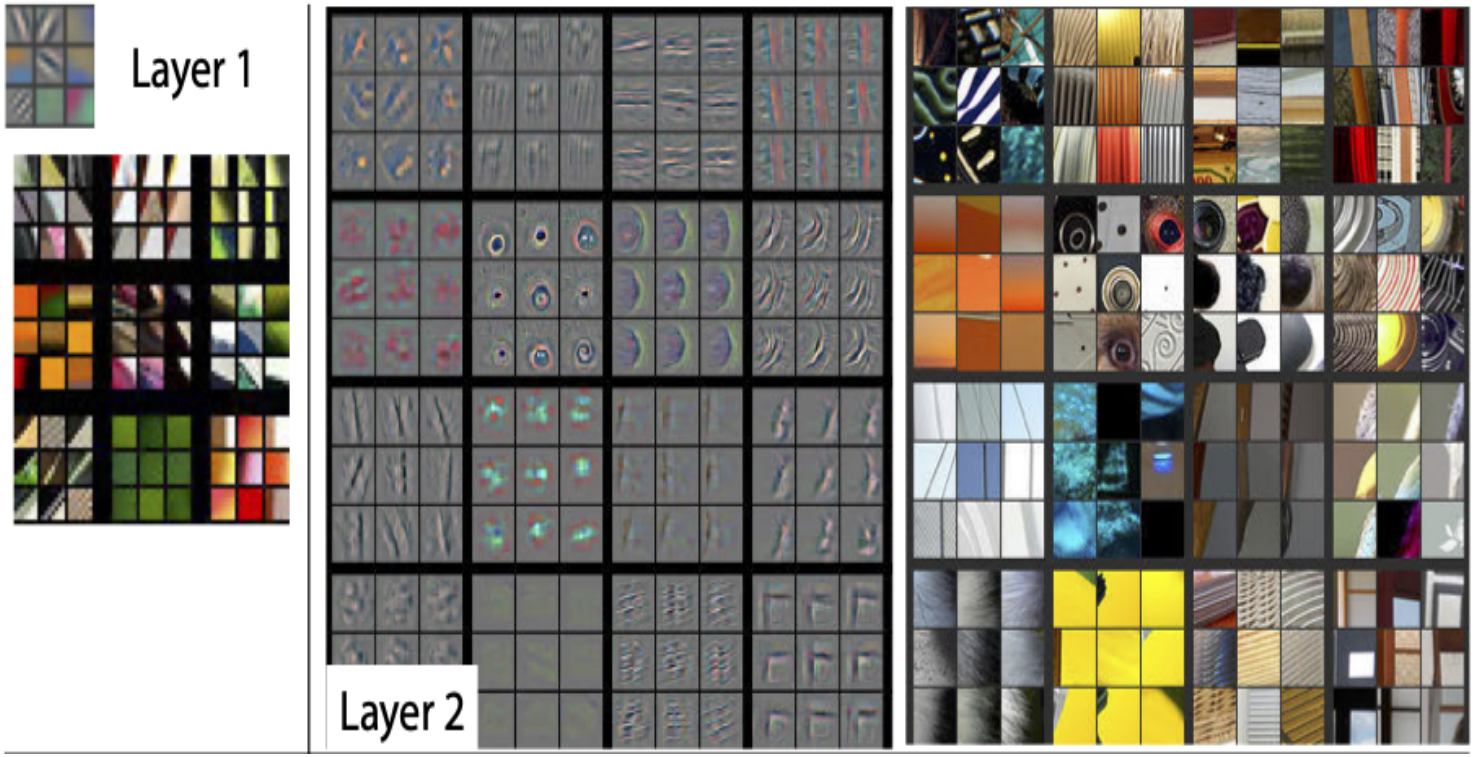
Figure 16: Activation maximization of the first filters of each convolutional layer in VGG-M.

Note, again we observe the same pattern, the initial filters are similar to Gabor/Wavelet filters and later layers are image components.

`https://arxiv.org/abs/1512.02017`

Deep CNN (Zeiler et al. 13')

Learned / memorized complex structure from data classes



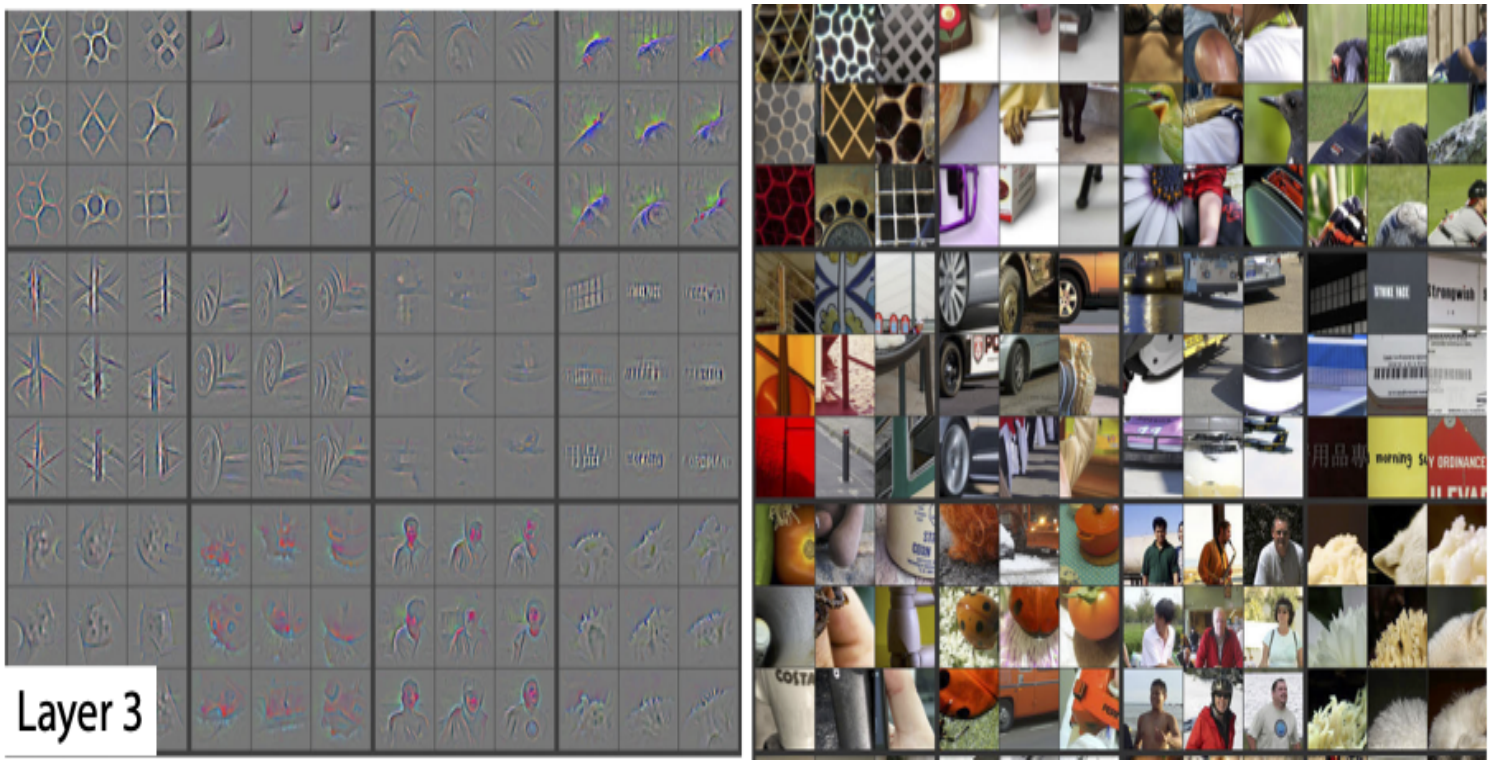
Layer 1 are masks, subsequent layers are their linear combinations.

`https://arxiv.org/abs/1311.2901`

Deep CNN (Zeiler et al. 13')

Learned / memorized complex structure from data classes

University of Oxford
Mathematical Institute



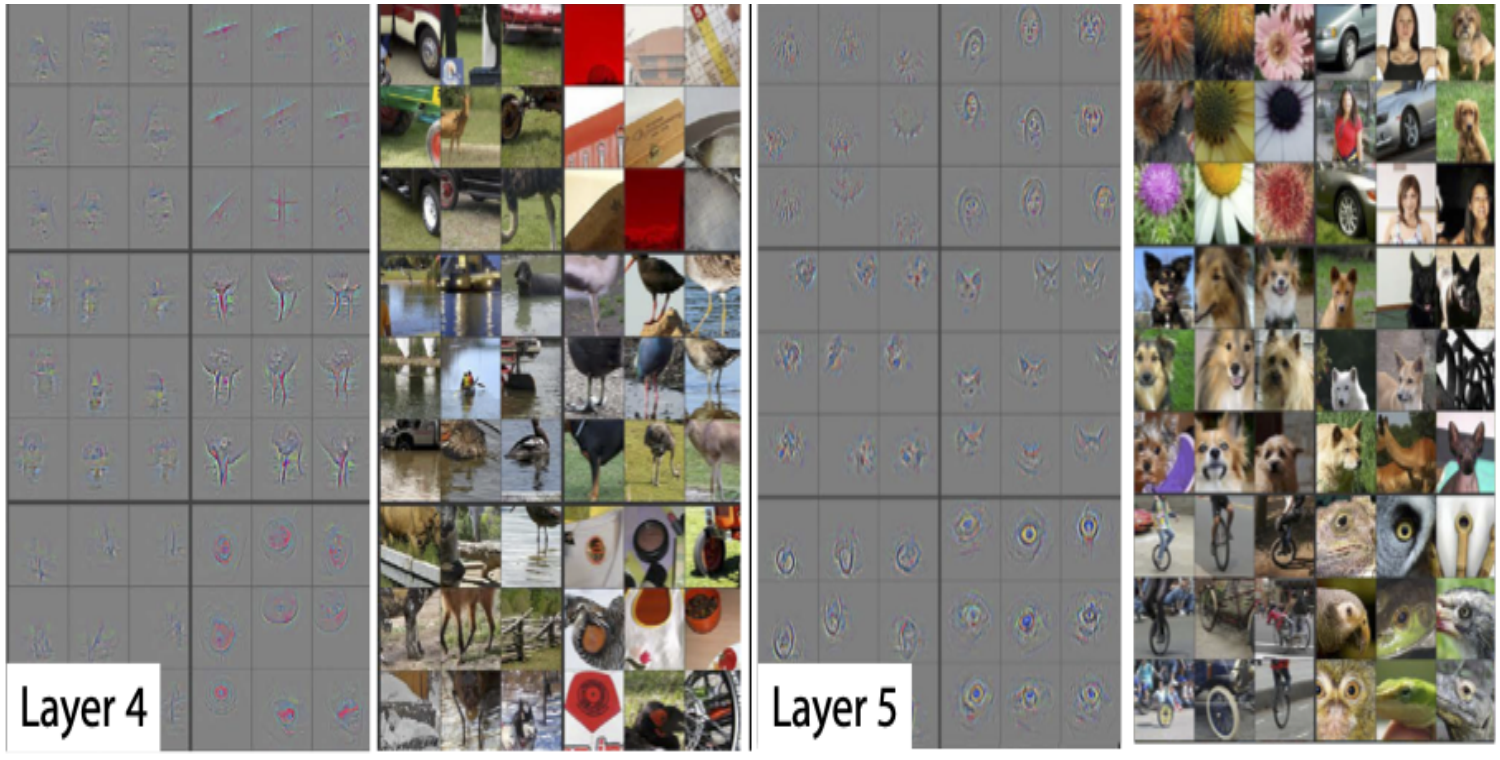
Layer 1 are masks, subsequent layers are their linear combinations.

`https://arxiv.org/abs/1311.2901`

Deep CNN (Zeiler et al. 13')

Learned / memorized complex structure from data classes

University of Oxford
Mathematical Institute



Layer 1 are masks, subsequent layers are their linear combinations.

`https://arxiv.org/abs/1311.2901`

Summary: similarity and importance of initial layers

Importance of training initial layers to develop representation

UNIVERSITY OF OXFORD
Mathematical Institute

We observe the initial layer of CNNs to be similar to one another, and to exhibit wavelet like representations. This is to be expected.

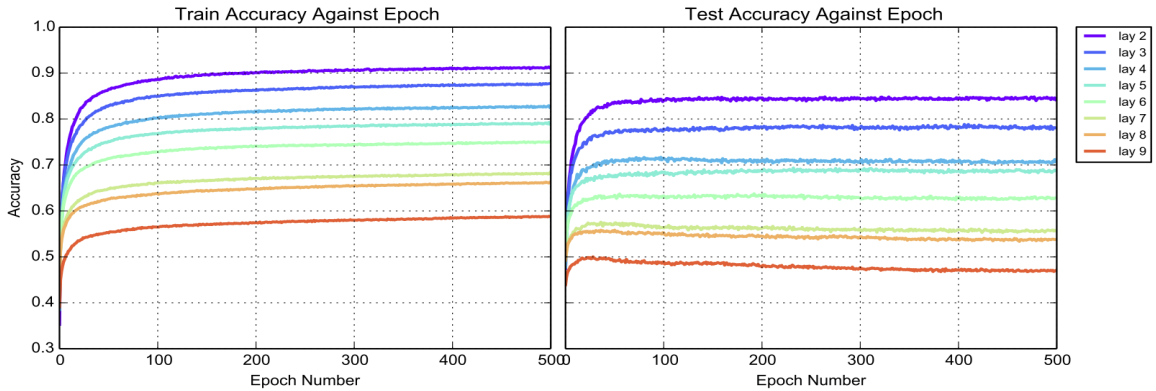


Figure 6: Demonstration of expressive power of remaining depth on MNIST. Here we plot train and test accuracy achieved by training exactly one layer of a fully connected neural net on MNIST. The different lines are generated by varying the hidden layer chosen to train. All other layers are kept frozen after random initialization.

Accuracy of a *random network* is improved most by training earlier layers (Raghu 16').

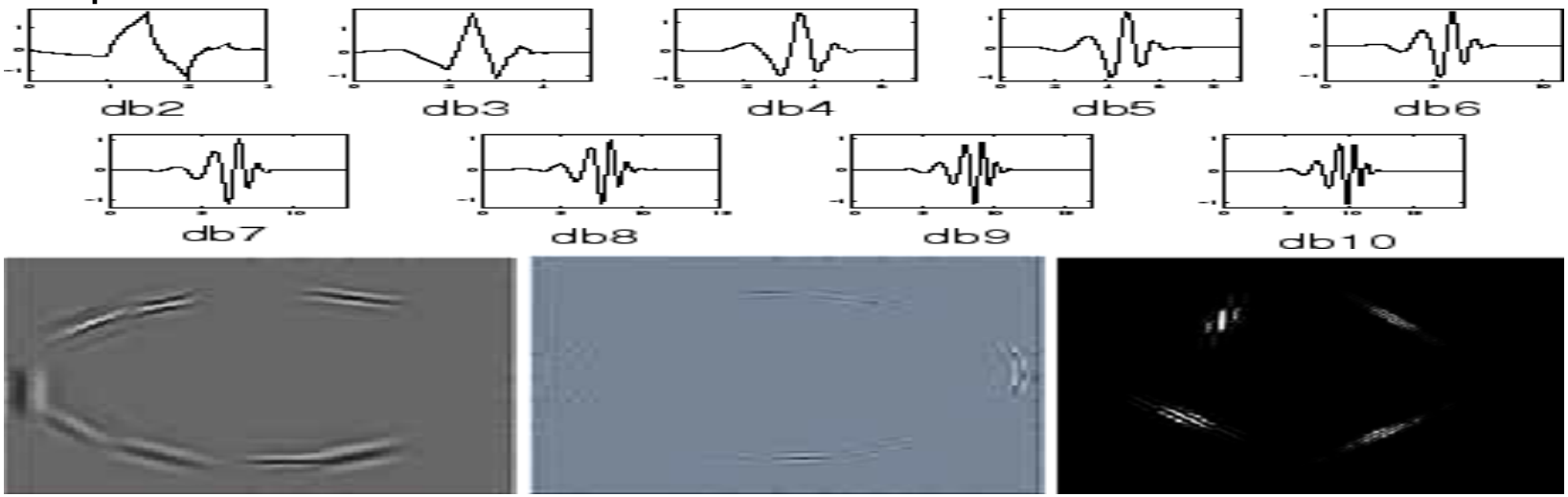
`https://arxiv.org/pdf/1611.08083.pdf`

Wavelet, curvelet, and contourlet: fixed representations

Known optimal representations for natural images

University of Oxford
Mathematical Institute

Applied and computational harmonic analysis community developed representations with optimal approximation properties for piecewise smooth functions.



Most notable are the Daubechies wavelets and Curvelets/Contourlets pioneered by Candes and Donoho.
While optimal, in a certain sense, for a specific class of functions, they can typically be improved upon for any particular data set.

Optimality of curvelets in 2D

Near optimality suggest a good initial CNN layer.

Mathematical Institute



Theorem (Candes and Donoho 02')

Let f be a two dimensional function that is piecewise C^2 with a boundary that is also C^2. Let f_n^F, f_n^W, and f_n^C be the best approximation of f using n terms of the Fourier, Wavelet and Curvelet representation respectively. Then their approximation error satisfy $\|f - f_n^F\|_{L^2}^2 = \mathcal{O}(n^{-1/2})$, $\|f - f_n^W\|_{L^2}^2 = \mathcal{O}(n^{-1})$, and $\|f - f_n^C\|_{L^2}^2 = \mathcal{O}(n^{-2}\log^3(n))$; moreover, no fixed representation can have a rate exceeding $\mathcal{O}(n^{-2})$.

`http://www.curvelet.org/papers/CurveEdges.pdf`

Initial layers can start as representations for the data class

Transfer learning: training only the final classification layer

UNIVERSITY OF OXFORD
Mathematical Institute

The first layer of a CNN can be initialized from a known representation for the data class. One can perform classification based on two layer net: layer 1: $h_2(x) = \sigma(W^{(1)}x + b^{(1)})$ where $W^{(1)}$ is a fixed transform of x to, say, the wavelet domain and $\sigma(\cdot)$ project to keep just the largest entries with hard or soft thresholding;

$$\sigma_{hard}(x;\tau) = \begin{cases} x & x > \tau \\ 0 & |x| \leq \tau \\ -x & x < -\tau \end{cases}, \quad \sigma_{soft}(x;\tau) = \begin{cases} x-\tau & x > \tau \\ 0 & |x| \leq \tau \\ -x+\tau & x < -\tau \end{cases}$$

layer2: $h_3 = \sigma(W^{(2)}h_2 + b^{(2)})$ with $W^{(2)}$ learned as the classifier based on the sparse codes h_2. However, h_2 does not build in invariance we would desire in classification, such as dilation, rotation, translation, etc... Depth remains important to generate these.

Individual neurons have associations with colour, materials, components, etc...

Understanding individual units in a DNN (Bau et al. 20')

Single units which reliably detect object classes

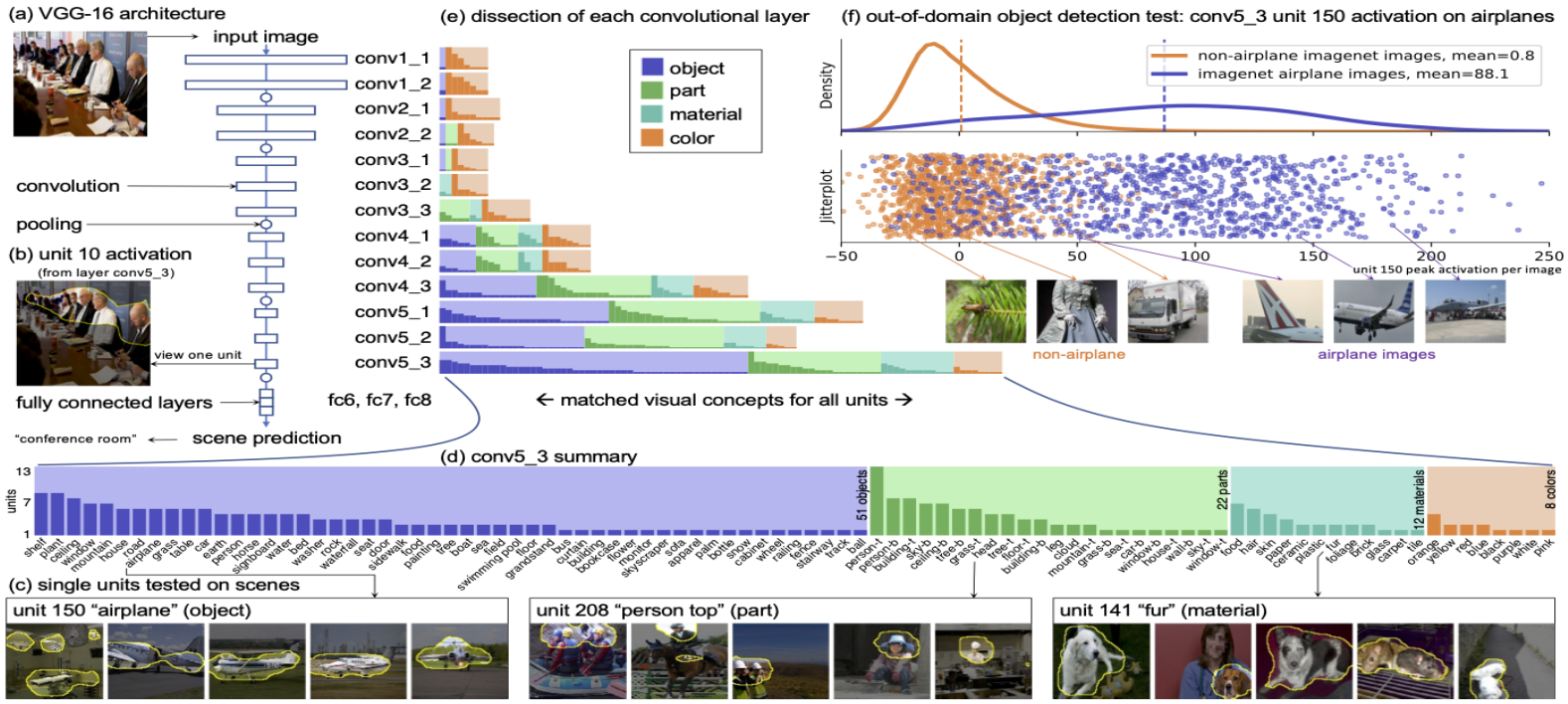


Fig. 1. The emergence of single-unit object detectors within a VGG-16 scene classifier. (a) VGG-16 consists of 13 convolutional layers, `conv1_1` through `conv5_3`, followed by three fully connected layers, `fc6,7,8`. (b) The activation of a single filter on an input image can be visualized as the region where the filter activates beyond its top 1% quantile level. (c) Single units are scored by matching high-activating regions against a set of human-interpretable visual concepts; each unit is labeled with its best-matching concept and visualized with maximally-activating images. (d) Concepts that match units in the final convolutional layer are summarized, showing a broad diversity of detectors for objects, object parts, materials, and colors. Many concepts are associated with multiple units. (e) Comparing all the layers of the network reveals that most object detectors emerge at the last convolutional layers. (f) Although the training set contains no object labels, unit 150 emerges as an 'airplane' object detector that activates much more strongly on airplane objects than non-airplane objects, as tested against a dataset of labeled object images not previously seen by the network. The jitter plot shows peak activations for the unit on randomly sampled 1,000 airplane and 1,000 non-airplane Imagenet images, and the curves show the kernel density estimates of these activations.

https://arxiv.org/abs/2009.05041

Input maximization and interpretability of LLMs

Maximization in word embedding (Poerner et al. 18')

Words or phrases that maximize outputs

UNIVERSITY OF OXFORD
Mathematical Institute

method	optimal 5-gram	target neuron activation
crp	pizza a sandwich and appetizers	48.44
gbl	fangs calzone raspberries sandwhich pizzas	64.46
	315th neuron in visual projection layer	
crp	fighter jet flying in formation	31.25
gbl	propelleor phrases jetliners treetops flight	37.82
	657th neuron in visual projection layer	
crp	a woman sitting under an	13.28
gbl	campbell lawn raincoat under an	17.54
	314th neuron ("umbrella") in language model projection layer	
crp	the view through a car	10.42
gbl	logging jeep watch through cracked	14.87
	957th neuron ("windshield") in language model projection layer	
crp	a giraffe looks to its	10.64
gbl	fest stares stares to their	13.22
	973th neuron ("left") in language model projection layer	

Table 1: Examples of optimal 5-grams via corpus search and via gradient ascent with gumbel softmax. Spelling errors stem from the Imaginet dictionary.

https://aclanthology.org/W18-5437.pdf

Role of transformer heads in MLP (Elhang et al. 21')

Trigrams

One of the core challenges of mechanistic interpretability is to make neural network parameters meaningful by contextualizing them (see discussion by Voss *et al.* in Visualizing Weights [12]). By multiplying out the OV and QK circuits, we've succeeded in doing this: the neural network parameters are now simple linear or bilinear functions on tokens. The QK circuit determines which "source" token the present "destination" token attends back to and copies information from, while the OV circuit describes what the resulting effect on the "out" predictions for the next token is. Together, the three tokens involved form a "skip-trigram" of the form `[source]... [destination][out]`, and the "out" is modified.

`https://transformer-circuits.pub/2021/framework/index.html`

Role of transformer heads in MLP (Elhang et al. 21')

Trigram example

UNIVERSITY OF OXFORD
Mathematical Institute

Some examples of large entries QK/OV circuit

Source Token	Destination Token	Out Token	Example Skip Tri-grams
" perfect"	" are", " looks", " is", " provides"	" perfect", " super", " absolute", " pure"	" perfect... are perfect", " perfect... looks super"
" large"	" contains", " using", " specify", " contain"	" large", " small", " very", " huge"	" large... using large", " large... contains small"
" two"	" One", "\n ", " has", "\r\n ", "One"	" two", " three", " four", " five", " one"	" two... One two", " two... has three"
"lambda"	" $\\", "}{\\", "+\\", "(\\", " ${\\"	"lambda", "sorted", " lambda", "operator"	"lambda... $\\lambda", "lambda... +\\lambda"
"nbsp"	"&", " \"&", "}&", ">&", "=&"	"nbsp", "01", "gt", "00012", "nbs", "quot"	"nbsp... ", "nbsp... > "
"Great"	"The", " The", " the", " contains", " /"	" Great", " great", " poor", " Every"	"Great... The Great", "Great... the great"

In the above example, we fix a given source token and look at the largest corresponding QK entries (the destination token) and largest corresponding OV entries (the out token). The source token is selected to show interesting behavior, but the destination and out token are the top entries unless entries are explicitly skipped with an ellipsis; they are colored by the intensity of their value in the matrix.

https://transformer-circuits.pub/2021/framework/index.html

Attention Sinks: structure in the attention matrices

Attention mechanism (Vaswani 17'), equations

Key and Query quadratic form to highlight relations

The input matrix X is then mapped to a query, key, and value matrices: $Q = XW_Q$, $K = XW_K$, and $V = XW_V$. The attention matrix

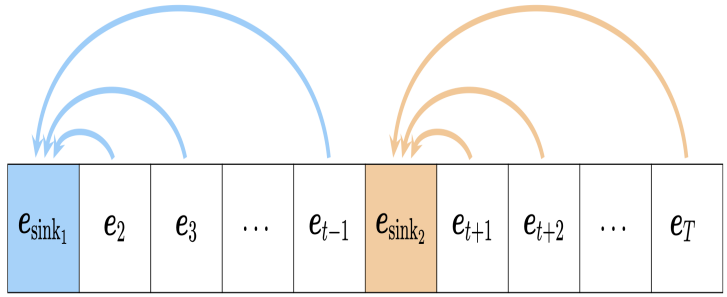
$$A = \text{softmax}\left(\frac{XW_Q W_K^T X^T}{n^{1/2}}\right)$$

has non-negative entries with each column summing to one. Generally Q and K have layer-norm applied to enforce fixed mean and variance. Intuitively the softmax helps highlight the rows in X that deserve "attention."

Attention Sinks (Zhang et al '25)

Structure we observed in softmax attention layers: catch 1

University of Oxford
Mathematical Institute



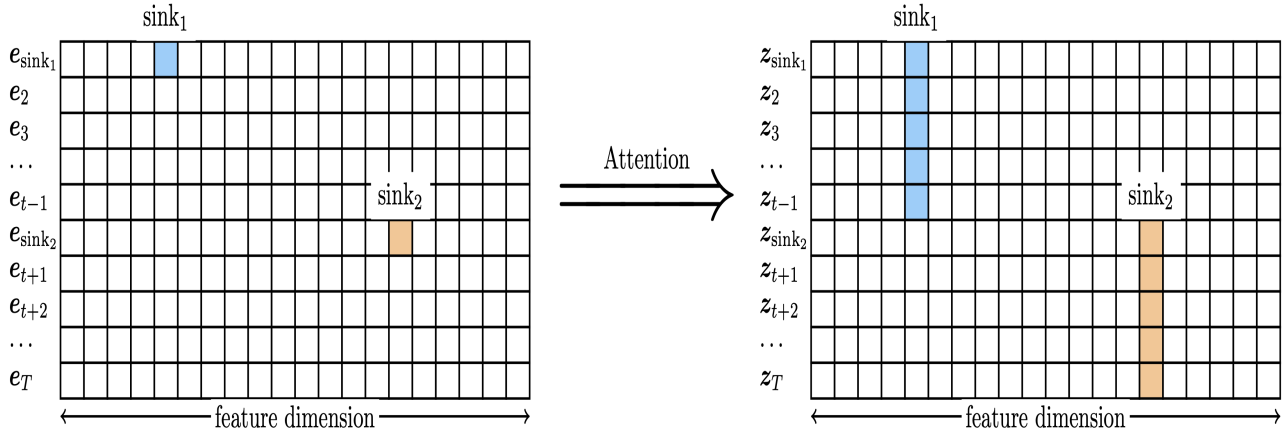
(a) **Catch**: Each box corresponds to a different token at the *input of the attention layer*, whose activation is denoted by e_i, and the arrows represent attention interactions. The *attention sinks* e_{sink_1} and e_{sink_2} *catch* the attention of tokens $e_2, e_3, \ldots, e_{t-1}$ and $e_{t+1}, e_{t+2}, \ldots, e_T$, respectively. This causes vertical bands to emerge in the attention weights A, as shown in Figure 2a.

`https://arxiv.org/abs/2502.00919`

Attention Sinks (Zhang et al '25)

Structure we observed in softmax attention layers: catch 2

UNIVERSITY OF OXFORD
Mathematical Institute



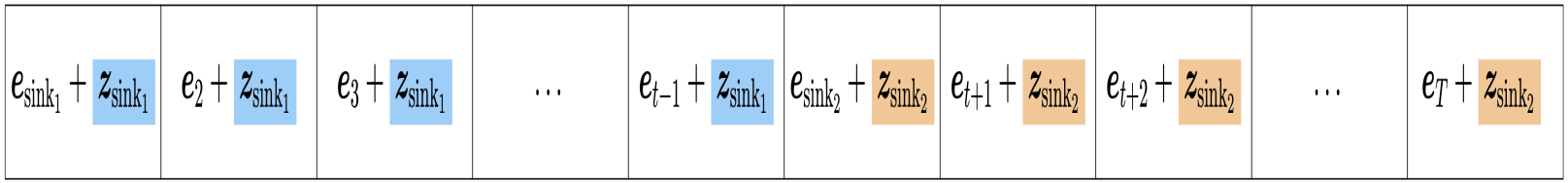
(b) **Tag:** The left grid shows the *attention value matrix* $\boldsymbol{V} = [e_1; \ldots; e_T]$, where activation vectors e_i are stacked vertically. The right grid shows the output of the attention layer $\boldsymbol{Z} = [z_{\text{sink}_1}; z_2; \ldots; z_T] = \boldsymbol{AV}$, with output vectors z_i also stacked vertically. The value vectors of the sinks, e_{sink1} and e_{sink_2}, are copied to all tokens that attend to them, thereby tagging them. These tags cause the token representation to cluster based on the sink they attended to, as revealed in the PCA plot in Figure 2c. The inputs to the attention layer, prior to the tagging, show no such clustering, as shown in Figure 2b.

`https://arxiv.org/abs/2502.00919`

Attention Sinks (Zhang et al '25)

Structure we observed in softmax attention layers: release

UNIVERSITY OF OXFORD
Mathematical Institute



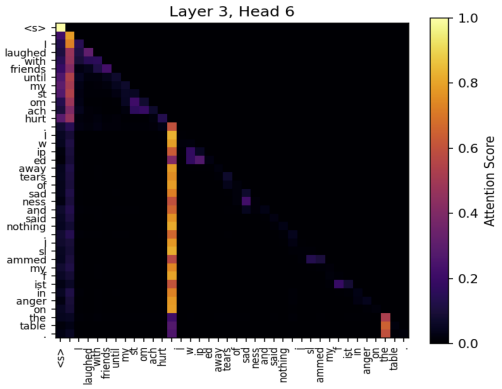
(c) **Release:** Each box corresponds to a different token at the *output of the attention layer*. The attention outputs are added to the residual stream as $e_i + z_{\text{sink}}$, creating common directions in representation space, in the form of the tags z_{sink_1} and z_{sink_2} shared across multiple tokens. These tags cause the token representations to cluster in deeper layers, as revealed in the PCA plot in Figure 2d.

`https://arxiv.org/abs/2502.00919`

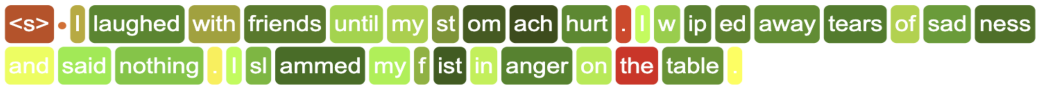
Attention Sinks (Zhang et al '25)

Structure we observed in softmax attention layers: clustering

UNIVERSITY OF OXFORD
Mathematical Institute



(a) **Attention Weights.** Two attention sinks catch the attention of subsequent tokens in the sequence.



(b) **PCA on input to the attention layer.** Tokens exhibit no clustering.



(c) **PCA on the output of the attention layer.** Tokens cluster according to their *attended sink*: those attending to the first sink are shaded red, while those attending to the second sink are shaded green.



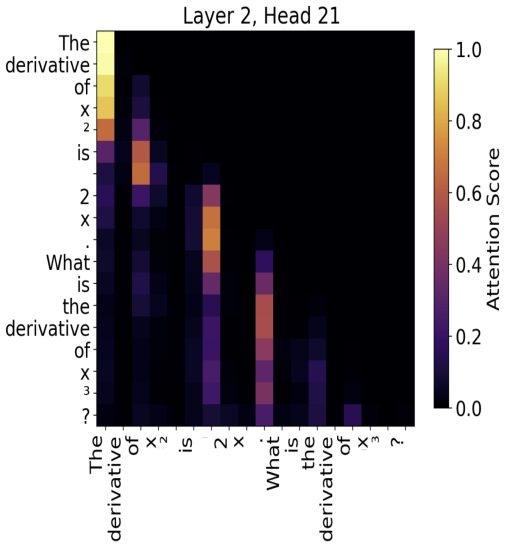
(d) **PCA on the residual stream in a deeper layer.** Tagged tokens propagate through the residual stream, clustering in a deeper layer based on their previously attended sink. Tokens that attended to the first sink are green, while those attending to the second sink are brown and yellow.

Figure 2: **Qualitative analysis of the 'catch, tag, release' mechanism.** The second and third subplots use PCA-based coloring of embeddings, described in Section 2. Appendix A presents additional measurements across a wide range of models, layers, attention heads, and prompts, including chain-of-thought [Wei et al., 2022], and zero-shot chain-of-thought [Brown et al., 2020].

https://arxiv.org/abs/2502.00919

Attention Sinks (Zhang et al '25)

Structure we observed in softmax attention layers: example



(a) Attention probabilities at layer 2, head 21.



(b) Input embedding visualization at layer 2.



(c) Attention head output at layer 2, head 21.



(d) PCA on residual stream at layer 38.

Figure 9: Visualization of the 'catch, tag, release' mechanism on a sample one-shot learning prompt on the QWEN 2.5-32B-INSTRUCT model.

https://arxiv.org/abs/2502.00919

The Scattering Transform: a deep transform framework with learning only in the last layer

Scattering Transform (Mallat 12')

Repeated application of deterministic transforms

University of Oxford
Mathematical Institute

The Scattering Transform repeatedly applied a deterministic wavelet transform followed by $\sigma(x) = |x|$ as nonlinear activation

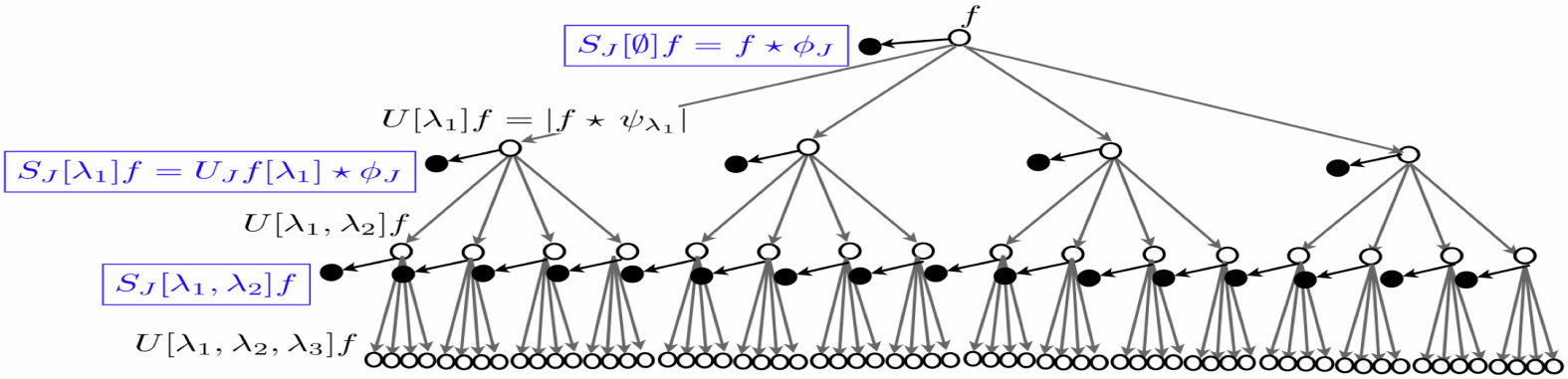


Figure 1: A scattering propagator U_J applied to f computes each $U[\lambda_1]f = |f \star \psi_{\lambda_1}|$ and outputs $S_J[\emptyset]f = f \star \phi_{2^J}$. Applying U_J to each $U[\lambda_1]f$ computes all $U[\lambda_1, \lambda_2]f$ and outputs $S_J[\lambda_1] = U[\lambda_1] \star \phi_{2^J}$. Applying iteratively U_J to each $U[p]f$ outputs $S_J[p]f = U[p]f \star \phi_{2^J}$ and computes the next path layer.

Depth allows the transform to become increasingly invariant to translation and small diffeomorphisms.

`https://arxiv.org/pdf/1101.2286.pdf`

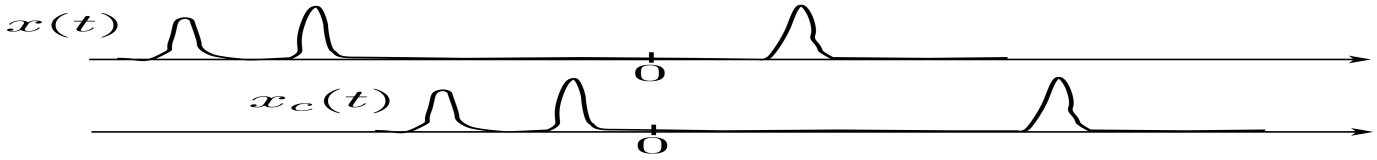
Classification as learning invariance (Mallat '13)

Projecting out invariants not needed for classification

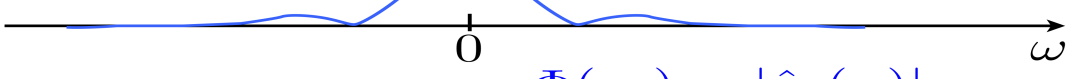
Mathematical Institute

Invariance to translations $x_c(t) = x(t - c)$

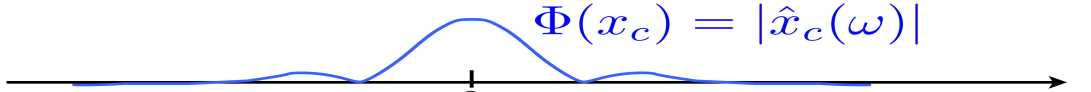
$$\forall c \in \mathbf{R} \quad , \quad \Phi(x_c) = \Phi(x) \ .$$



$\Phi(x) = |\hat{x}(\omega)|$: Fourier Modulus



$\Phi(x_c) = |\hat{x}_c(\omega)|$



Lipschitz stable to deformations $x_\tau(t) = x(t - \tau(t))$

small deformations of x $\Longrightarrow$ small modifications of $\Phi(x)$

$$\forall \tau \quad , \quad \|\Phi(x_\tau) - \Phi(x)\| \leq C \underbrace{\sup_t |\nabla \tau(t)|}_{\text{deformation size}} \|x\| \ .$$

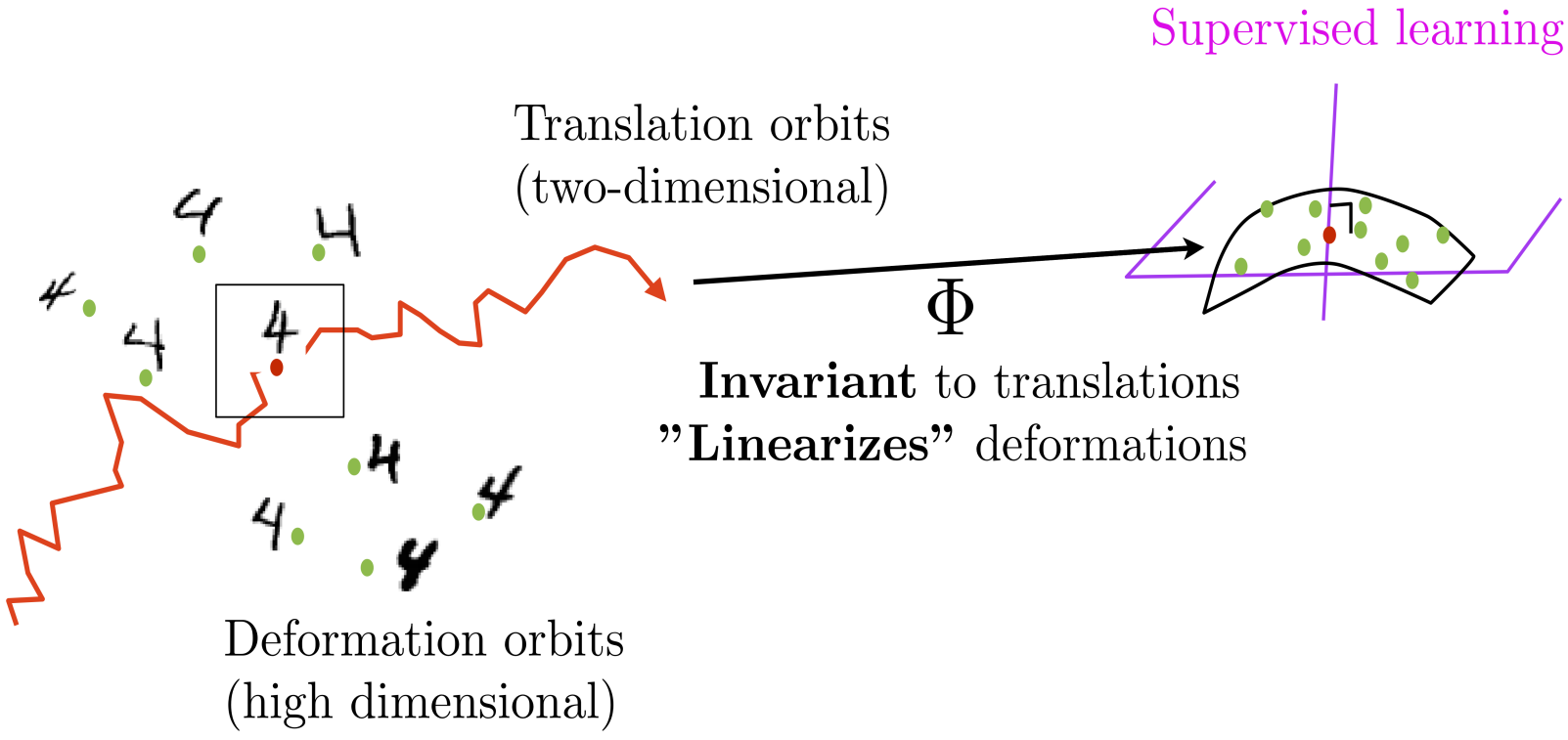
http://lcsl.mit.edu/ldr-workshop/Home.html

Linearising deformations (Mallat '13)

Projecting out invariants not needed for classification

Mathematical Institute

- Specific deformation invariance must be learned.

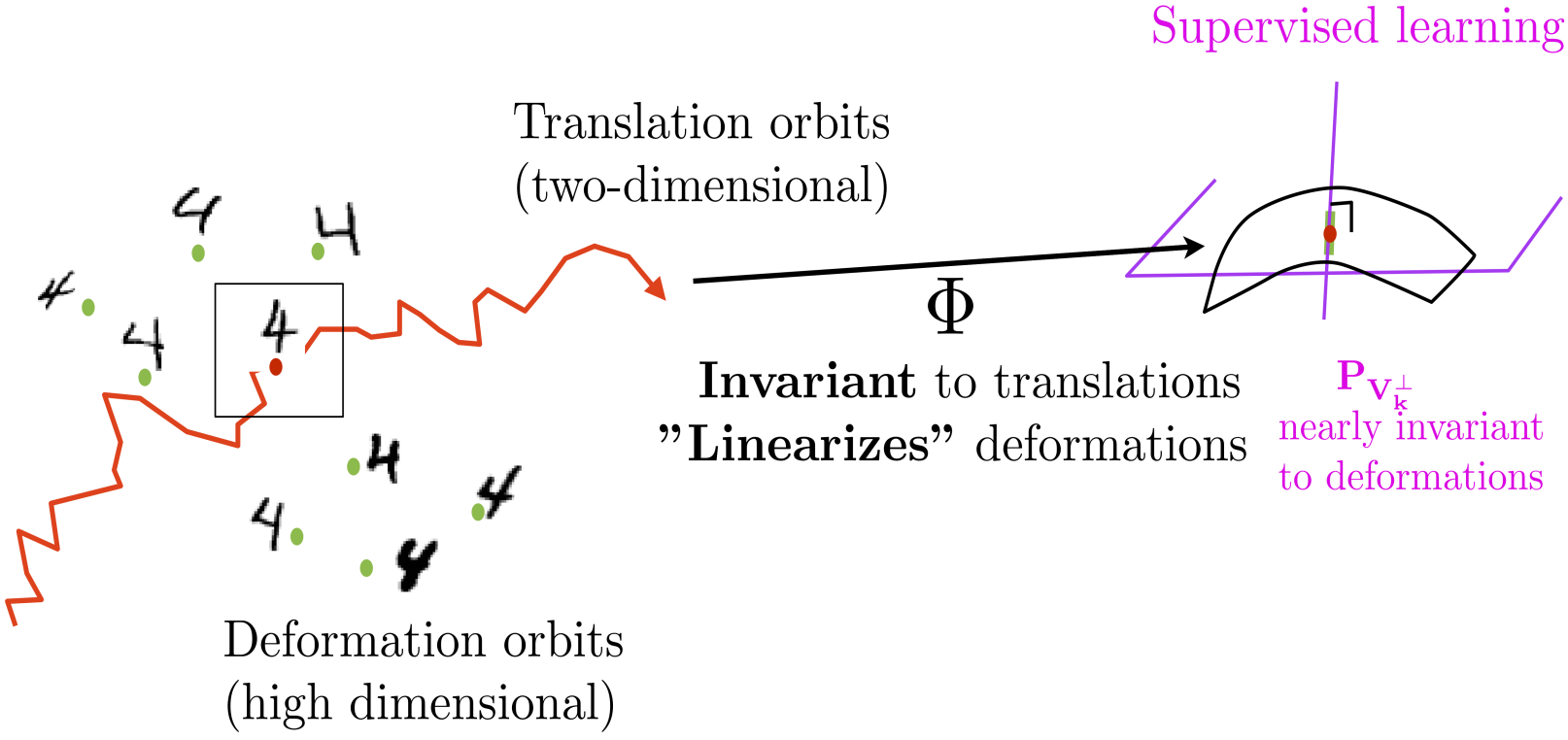


`http://lcsl.mit.edu/ldr-workshop/Home.html`

Linearising deformations (Mallat '13)

Projecting out invariants not needed for classification

- Specific deformation invariance must be learned.



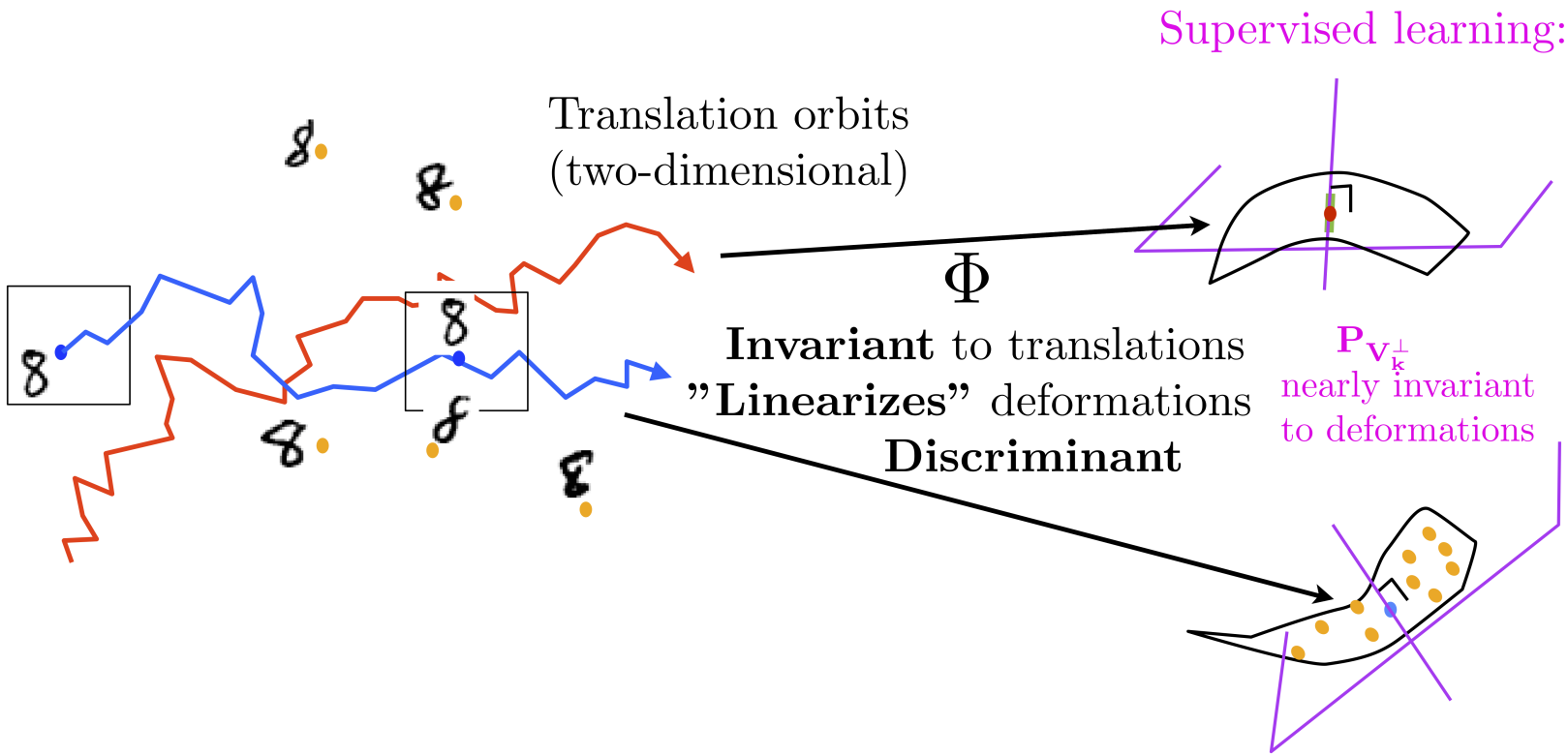
`http://lcsl.mit.edu/ldr-workshop/Home.html`

Linearising deformations (Mallat '13)

Projecting out invariants not needed for classification

UNIVERSITY OF OXFORD
Mathematical Institute

- Specific deformation invariance must be learned.

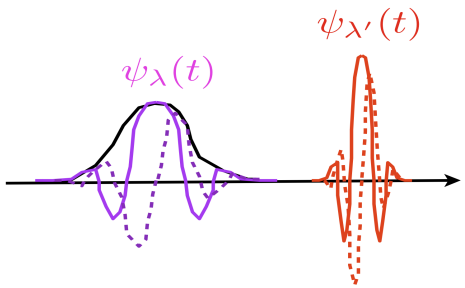


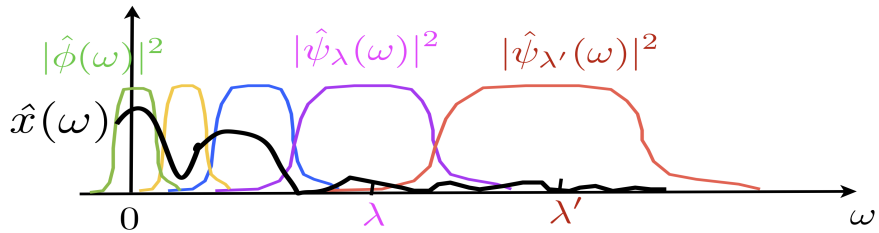
`http://lcsl.mit.edu/ldr-workshop/Home.html`

Wavelet Transform as frequency tiling (Mallat '13)

Wavelets decompose function into local frequency information

- Complex wavelet: $\psi(t) = \psi^a(t) + i\,\psi^b(t)$
- Dilated: $\psi_\lambda(t) = 2^{-j}\,\psi(2^{-j}t)$ with $\lambda = 2^{-j}$.





- Wavelet transform: $x \star \psi_\lambda(t) = \int x(u)\,\psi_\lambda(t-u)\,du$

$$Wx = \begin{pmatrix} x \star \phi(t) \\ x \star \psi_\lambda(t) \end{pmatrix}_{t,\lambda}$$

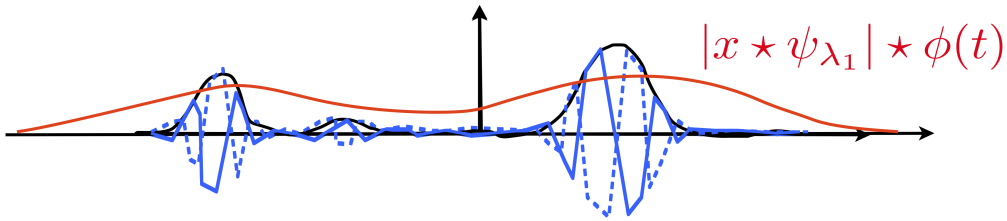
Unitary: $\|Wx\|^2 = \|x\|^2$.

http://lcsl.mit.edu/ldr-workshop/Home.html

Modulus and averaging in wavelet domain (Mallat '13)

Smoothing to identify discontinuities and have energy decay

University of Oxford
Mathematical Institute



- The modulus $|x \star \psi_{\lambda_1}|$ is a regular envelop
- The average $|x \star \psi_{\lambda_1}| \star \phi(t)$ is invariant to small translations relatively to the support of ϕ.
- Full translation invariance at the limit:

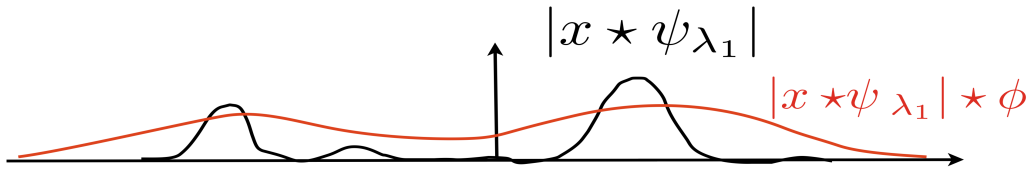
$$\lim_{\phi \to 1} |x \star \psi_{\lambda_1}| \star \phi(t) = \int |x \star \psi_{\lambda_1}(u)| \, du = \|x \star \psi_{\lambda_1}\|_1$$

`http://lcsl.mit.edu/ldr-workshop/Home.html`

Second layer of the scattering transform (Mallat '13)

Increased smoothness with depth

Mathematical Institute



- The high frequencies of $|x \star \psi_{\lambda_1}|$ are in wavelet coefficients:

$$W|x \star \psi_{\lambda_1}| = \begin{pmatrix} |x \star \psi_{\lambda_1}| \star \phi(t) \\ |x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}(t) \end{pmatrix}_{t,\lambda_2}$$

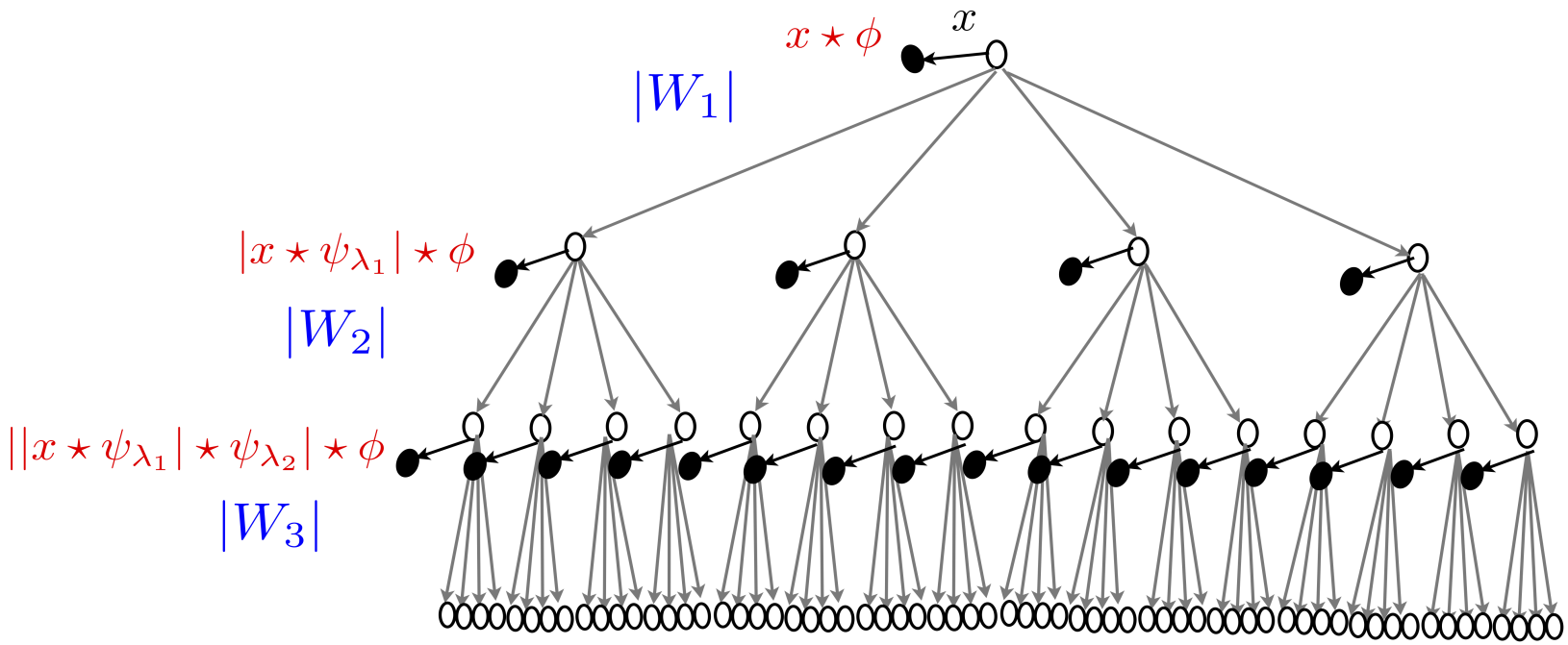
- Translation invariance by time averaging the amplitude:

$$\forall \lambda_1, \lambda_2, \quad ||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \star \phi(t)$$

`http://lcsl.mit.edu/ldr-workshop/Home.html`

Scattering transform (Mallat '13)

Lipshitz continuous, inputs contract to one another



- Cascade of contractive operators

$$\| |W_k|x - |W_k|x' \| \leq \|x - x'\| \quad \text{with} \quad \| |W_k|x \| = \|x\| \ .$$

`http://lcsl.mit.edu/ldr-workshop/Home.html`

Scattering transform properties(Mallat '13)

Stability to deformormations

$$Sx = \begin{pmatrix} x \star \phi\,(u) \\ |x \star \psi_{\lambda_1}| \star \phi(u) \\ ||x \star \psi_{\lambda_1}| \star \psi_{\lambda_2}| \star \phi(u) \\ |||x \star \psi_{\lambda_2}| \star \psi_{\lambda_2}| \star \psi_{\lambda_3}| \star \phi(u) \\ \cdots \end{pmatrix}_{u,\lambda_1,\lambda_2,\lambda_3,\ldots}$$

Theorem: *For appropriate wavelets, a scattering is*

contractive $\|Sx - Sy\| \leq \|x - y\|$

preserves norms $\|Sx\| = \|x\|$

stable to deformations $x_\tau(t) = x(t - \tau(t))$

$$\|Sx - Sx_\tau\| \leq C \sup_t |\nabla \tau(t)|\, \|x\|$$

$\Rightarrow$ linear discriminative classification from $\Phi x = Sx$

`http://lcsl.mit.edu/ldr-workshop/Home.html`

Scattering Transform: energy decay (Mallat 12')

The transform can be truncated stably

UNIVERSITY OF OXFORD
Mathematical Institute

Lemma

For suitably chosen wavelet transforms (see Theorem 2.6 in footnote) then for all $f \in L^2(\mathbb{R}^d)$

$$\lim_{m\to\infty} \|U[\Lambda_J^m]f\|^2 = \lim_{m\to\infty} \sum_{n=m}^{\infty} \|S_J[\Lambda_J^n]f\|^2 = 0$$

where $U[\lambda]f = |f \star \psi_\lambda|$ and $S_J[\lambda]f = \phi_j \star U[\lambda]f$ and $\|S_J[P_J]f\| = \|f\|$. Morevover, for all $c \in \mathbb{R}^d$

$$\lim_{J\to\infty} \|S_J[P_J]f - S_J[P_J]L_c f\| = 0$$

where $L_c f = f(x - c)$ is the translation operator.

`https://arxiv.org/pdf/1101.2286.pdf`

Scattering Transform: energy decay (Mallat 13')

Energy decay on CalTech-101

University of Oxford
Mathematical Institute

TABLE 1
Percentage of Energy $\sum_{p \in \mathcal{P}^m} \|S[p]x\|^2 / \|x\|^2$ of Scattering Coefficients on Frequency-Decreasing Paths of Length m, Depending upon J

J	$m=0$	$m=1$	$m=2$	$m=3$	$m=4$	$m \leq 3$
1	95.1	4.86	-	-	-	99.96
2	87.56	11.97	0.35	-	-	99.89
3	76.29	21.92	1.54	0.02	-	99.78
4	61.52	33.87	4.05	0.16	0	99.61
5	44.6	45.26	8.9	0.61	0.01	99.37
6	26.15	57.02	14.4	1.54	0.07	99.1
7	0	73.37	21.98	3.56	0.25	98.91

These average values are computed on the Caltech-101 database, with zero mean and unit variance images.

https://www.di.ens.fr/data/publications/papers/pami-final.pdf

Scattering Transform: MNIST classification (Mallat 13')

Accuracy on MNIST based on training size

UNIVERSITY OF OXFORD
Mathematical Institute

TABLE 4
Percentage of Errors of MNIST Classifiers, Depending on the Training Size

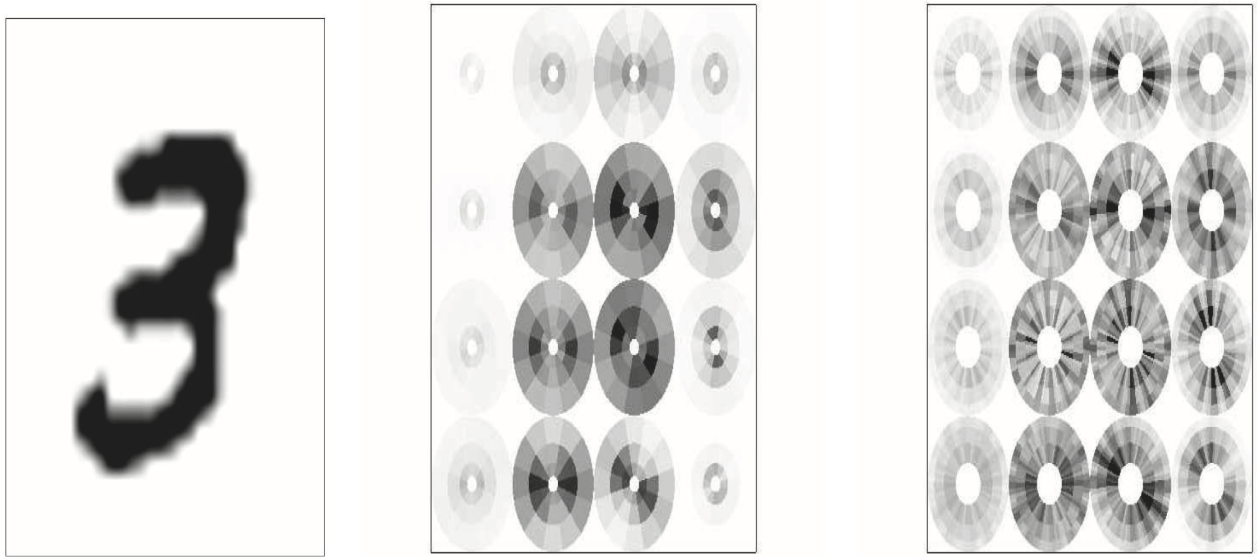
Training size	x PCA	x SVM	Wind. Four. PCA	Wind. Four. SVM	Scat. $\overline{m}=1$ PCA	Scat. $\overline{m}=1$ SVM	Scat. $\overline{m}=2$ PCA	Scat. $\overline{m}=2$ SVM	Conv. Net.
300	14.5	15.4	7.35	7.4	5.7	8	**4.7**	5.6	7.18
1000	7.2	8.2	3.74	3.74	2.35	4	**2.3**	2.6	3.21
2000	5.8	6.5	2.99	2.9	1.7	2.6	**1.3**	1.8	2.53
5000	4.9	4	2.34	2.2	1.6	1.6	**1.03**	1.4	1.52
10000	4.55	3.11	2.24	1.65	1.5	1.23	0.88	1	**0.85**
20000	4.25	2.2	1.92	1.15	1.4	0.96	0.79	**0.58**	0.76
40000	4.1	1.7	1.85	0.9	1.36	0.75	0.74	**0.53**	0.65
60000	4.3	1.4	1.80	0.8	1.34	0.62	0.7	**0.43**	0.53

`https://www.di.ens.fr/data/publications/papers/pami-final.pdf`

Scattering Transform: MNIST digit 3 (Mallat 13')

Example of energy in a scattering transform

University of Oxford
Mathematical Institute



(a) (b) (c)

Fig. 7. (a) Image $X(u)$ of a digit "3." (b) Arrays of windowed scattering coefficients $S[p]X(u)$ of order $m = 1$, with u sampled at intervals of $2^J = 8$ pixels. (c) Windowed scattering coefficients $S[p]X(u)$ of order $m = 2$.

https://www.di.ens.fr/data/publications/papers/pami-final.pdf