technische universität
dortmund

fakultät für
informatik

Deep Attribute Learning

Gernot A. Fink

TU Dortmund University, Department of Computer Science

2nd IWPAA-2018

In celebration of the 125th Birth Anniversary of Prof. P. C. Mahalanobis

- Attributes in Pattern Recognition — *Why are they useful?*
- Deep Learning — *How does it work?*
- Applications

  Scene Recognition, Word Spotting, Activity Recognition
- Conclusion

Joint work with: *René Grzeszick, Fernando Moya, Sebastian Sudholt*

Attributes

Semantic Attributes

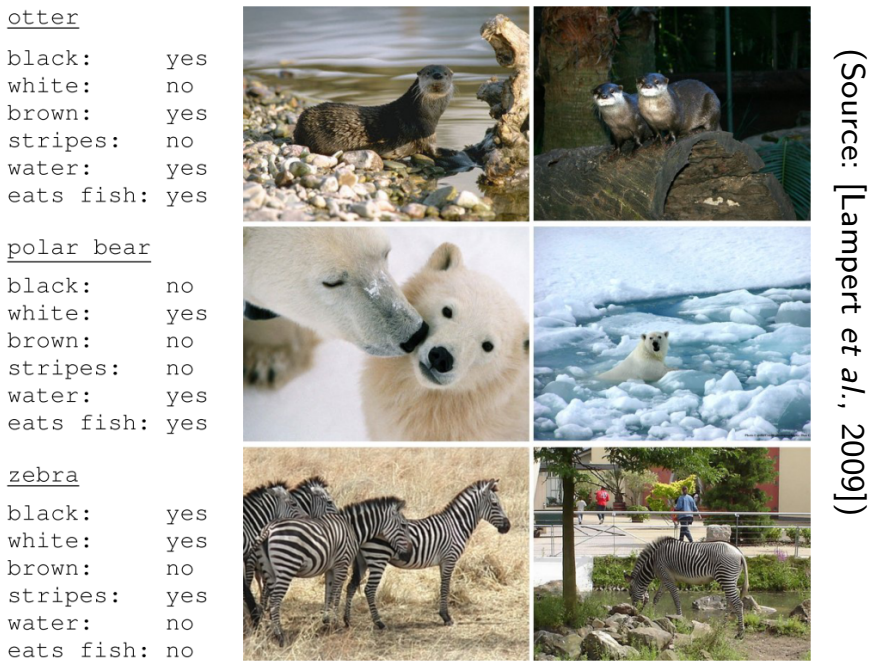
Goal: Perform *more robust* or even *zero-shot* classification

Idea: Describe classes by *shared* attributes

Specific configuration encodes a class

(Zero-Shot) Classification:

Assign new image to class y_i for which attribute representation matches best

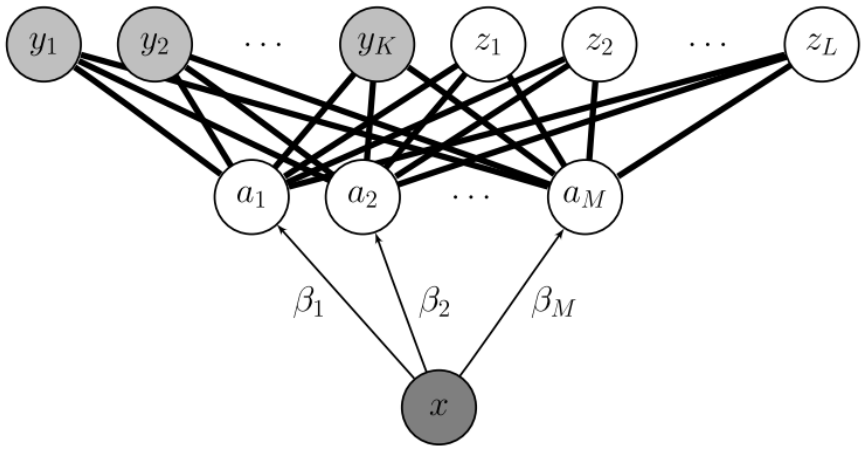


(Source: [Lampert *et al.*, 2009])

C. H. Lampert, H. Nickisch, S. Harmeling: Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer, Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, 951–958, 2009.
A. Farhadi, I. Endres, D. Hoiem, D. Forsyth: Describing objects by their attributes, Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, 1778–1785, 2009.

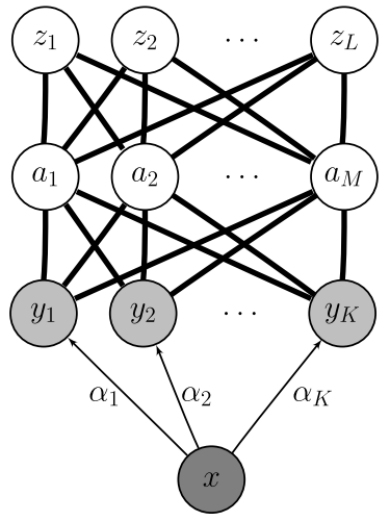
Direct/Indirect Attribute Prediction

Direct Attribute Prediction



1. Predict attribute a_i with classifier b_i from x
2. Use $\boldsymbol{a}$ to classify into either a known class y_i or a class z_i not seen in training

Indirect Attribute Prediction



(Source: [Lampert *et al.*, 2009])

1. Predict probability α_i of x belonging to class y_i
2. Compute weighted comb. $\boldsymbol{a}$ of attribute vectors of all y_i
3. Use $\boldsymbol{a}$ for classification

C. H. Lampert, H. Nickisch, S. Harmeling: Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer, Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, 951–958, 2009.

Classification with DAP

Input: Attribute vector $\boldsymbol{a}^{(l)}$ for each of the K classes (known and unknown)

Attribute classifiers predicting an attribute representation $\hat{\boldsymbol{a}}$ for $\boldsymbol{x}$

Output: Predicted class $\hat{z}$

Method (distance based):

$$\hat{z} = \underset{k=1\ldots K}{\operatorname{argmin}}\, d(\hat{\boldsymbol{a}}, \boldsymbol{a}_k)$$

Method (probabilistic) [Lampert *et al.* 2009]

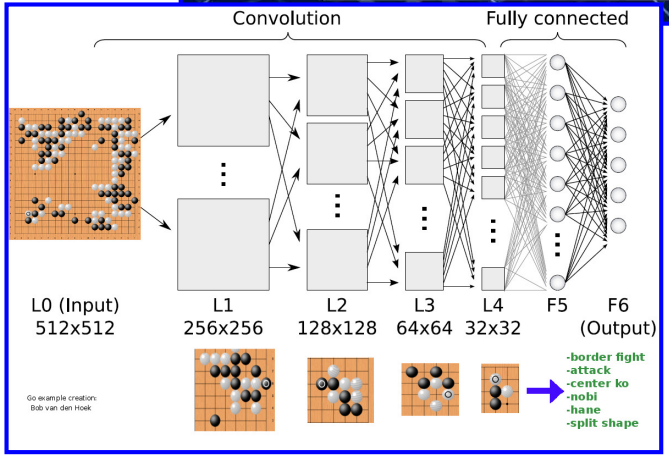
$$\hat{z} = \underset{k=1\ldots K}{\operatorname{argmax}}\, p(\hat{\boldsymbol{a}} = \boldsymbol{a}_k | \boldsymbol{x})$$

C. H. Lampert, H. Nickisch, S. Harmeling: Learning To Detect Unseen Object Classes by Between-Class Attribute Transfer, Proc. IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition, 951–958, 2009.

Deep Learning

The Deep Learning Hype

AlphaGo defeats Go-Master Lee Sedool 2016!



(Image sources: blogspot.com, go-baduk-weiqi.de, Nature)

⇒ *The Deep Learning Hype reaches the broader public!*

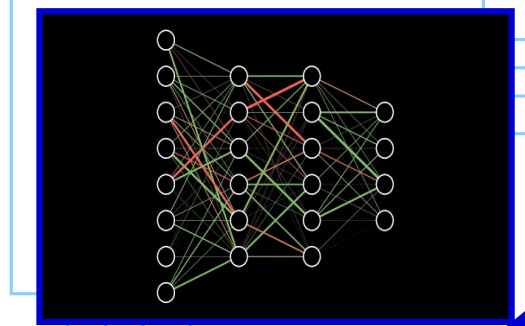
Deep Learning: Thematic Classification

Pattern Recognition

Tool Box

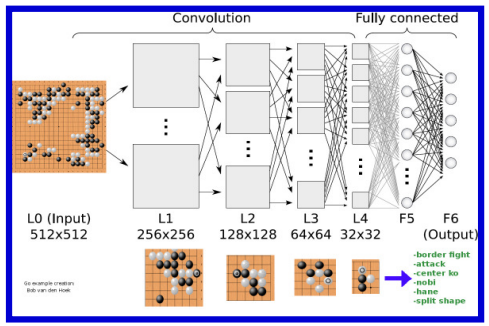
Machine Learning

Neural Networks



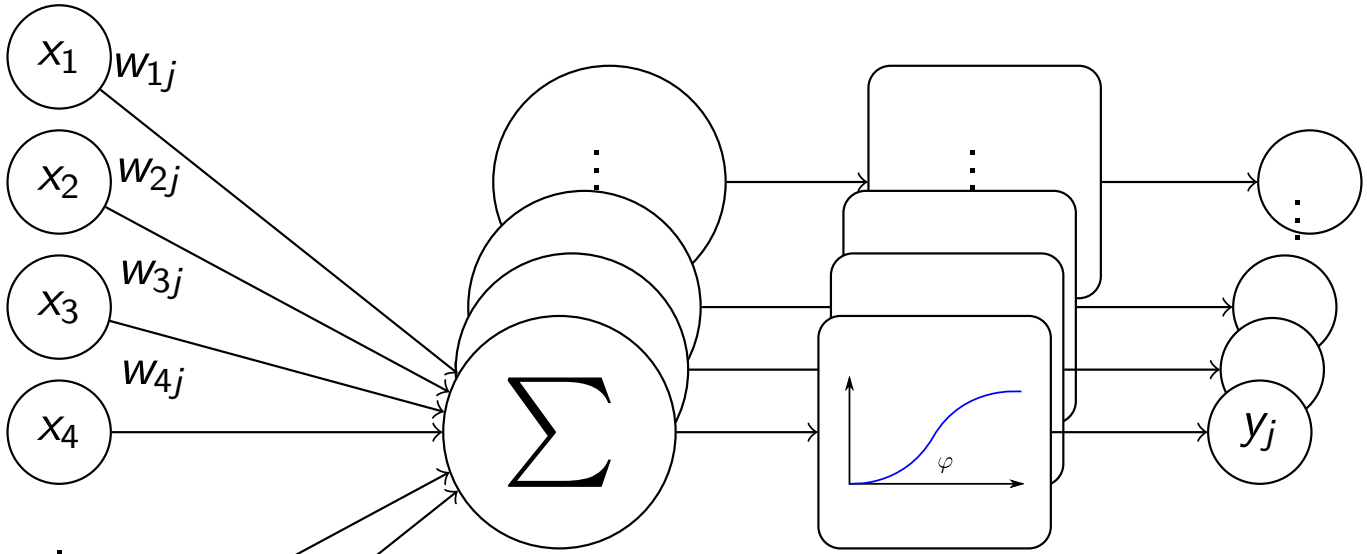
Specialization

Deep Learning



(Image sources: blog.finjan.com, tomtunguz.com, 3Blue1Brown @ youtube.com, blogspot.com)

The Perceptron



Single Output

$$y_j = \varphi\left(\sum_i w_{ij} x_i + b_j\right)$$

$$\mathbb{R}^n \rightarrow \mathbb{R}^m$$

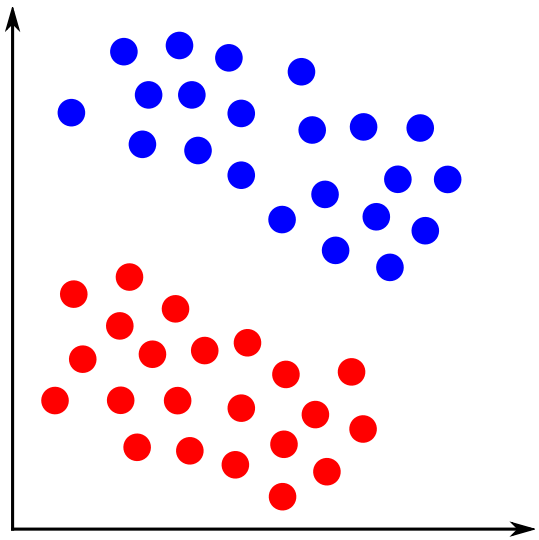
$$\mathbf{y} = \varphi\left(\mathbf{W}\mathbf{x} + \mathbf{b}\right)$$

F. Rosenblatt: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain, Psychological Review, 65(6), 1958.

Capabilities of the Perceptron

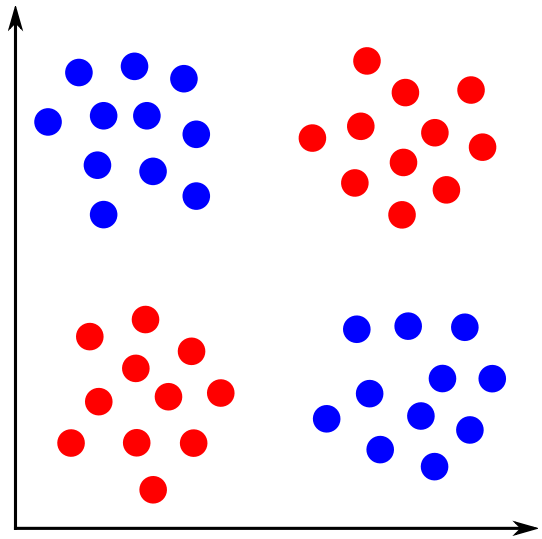
What a Perceptron can do:

Calssify two linearly separable classes



What a Perceptron can't do:

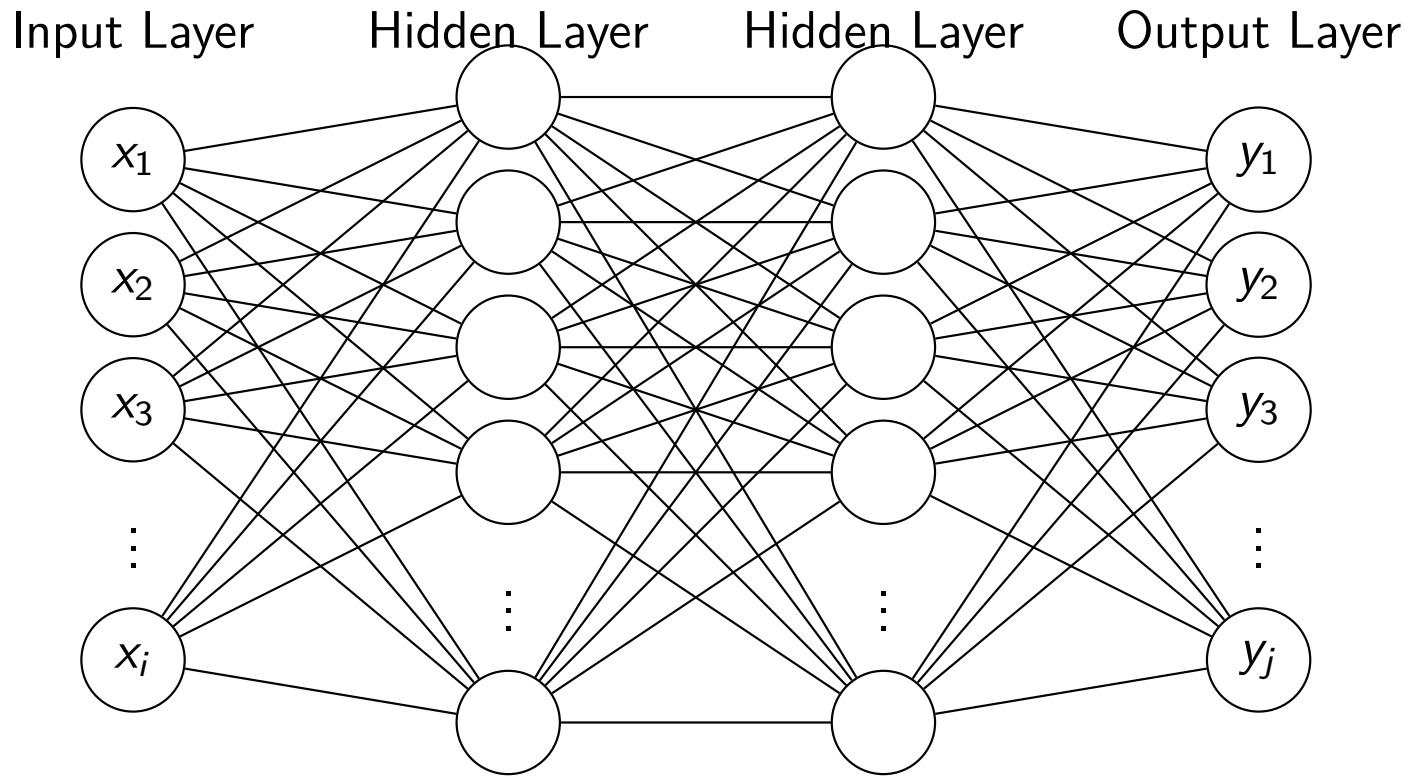
Classify two non-linearly separable classes (XOR-Problem)



Solution: Stack layers of Perceptrons

$\Rightarrow$ Multi Layer Perceptron

Multi Layer Perceptron (MLP)

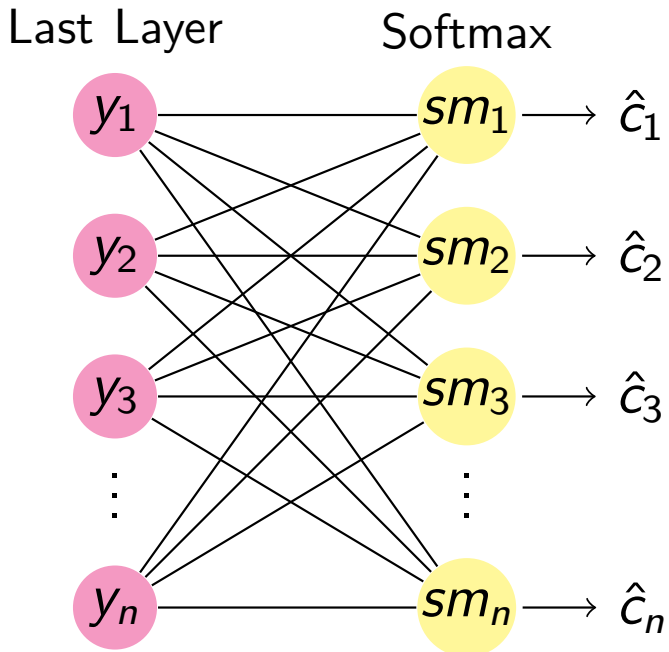


$$\mathbf{y} = \boldsymbol{f}^L\left(\boldsymbol{f}^{L-1}(\ldots \boldsymbol{f}^2\left(\boldsymbol{f}^1(\mathbf{x})\right)\right)$$

here: $$\mathbf{y} = \mathbf{W}_{\text{out}} \cdot \varphi\left(\mathbf{W}_{h2} \cdot \varphi\left(\mathbf{W}_{h1}\mathbf{x} + b_{h1}\right) + b_{h2}\right) + b_{\text{out}}$$

Classifying with MLPs

- For classification, the output of the MLP is *usually* forwarded through a Softmax Function: $sm_i(\mathbf{y}) = \frac{e^{y_i}}{\sum_j e^{y_j}}$
- sm_i can be interpreted as posterior probability for class c_i
- Predicted class: $\hat{c} = \max_i sm_i$

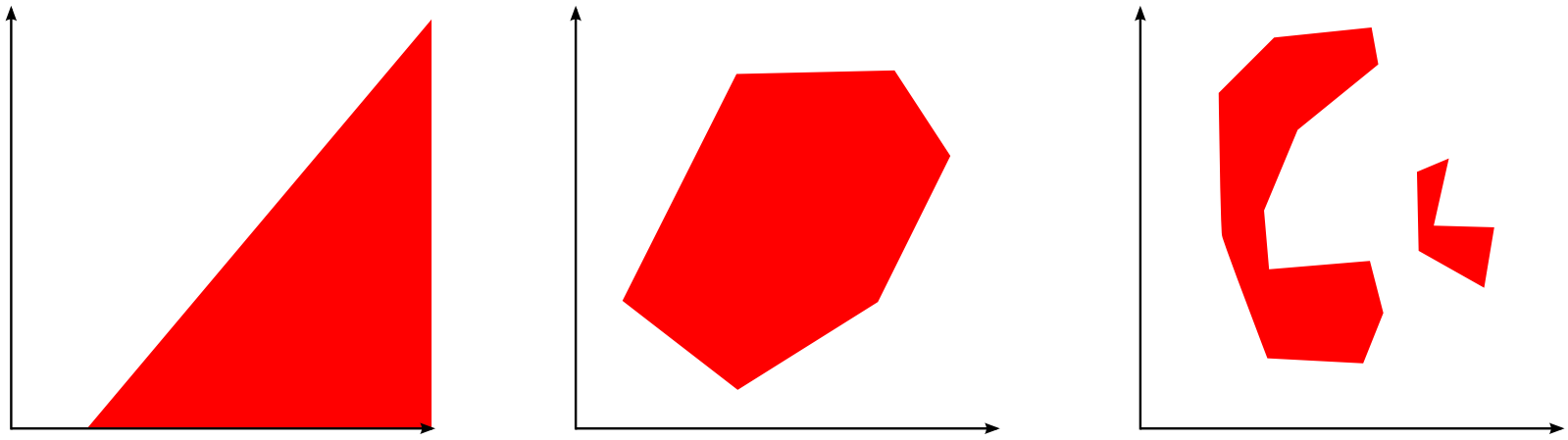


What an MLP Can Do!

... approximate any function (even with only 2 layers!)

[Hornik *et al.* 1989]

Interpretation with 3 layers (2 hidden, 1 output):



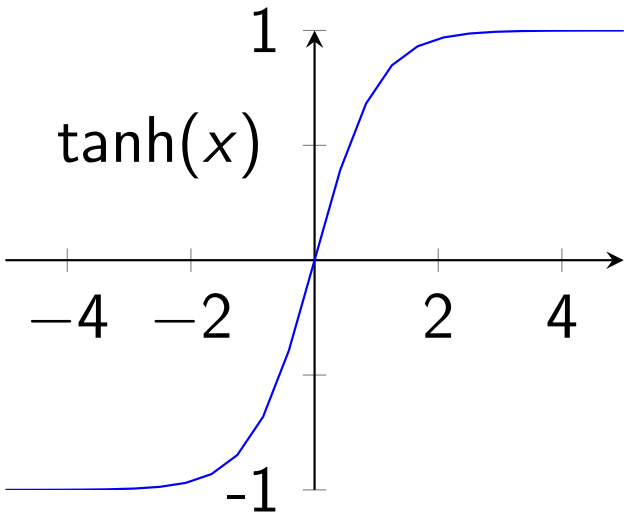
1. Layer: Halfspaces
2. Layer: Convex polyhedron
3. Layer: Multiple non-convex, non-connected polyhedra

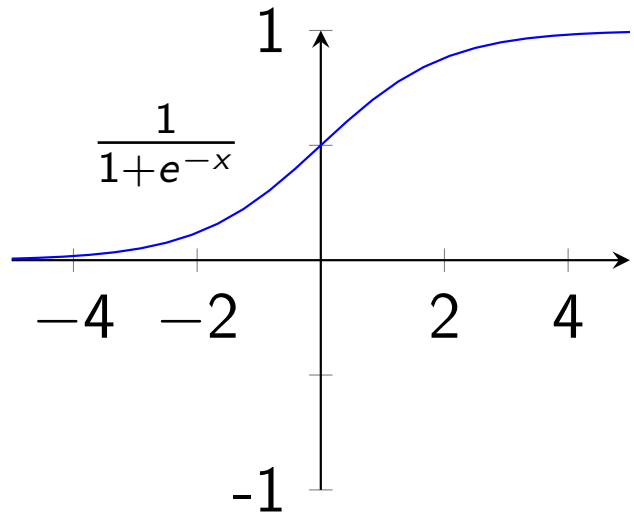
A Word on Activation Functions

- Activation functions are crucial for MLP
- Without non-linearities, an MLP implements a linear transform:

$$\mathbf{y} = \mathbf{W}_L \mathbf{W}_{L-1} \ldots \mathbf{W}_2 \mathbf{W}_1 \mathbf{x} = \mathbf{W}' \mathbf{x}$$

Classic Activation Functions: sigmoidal shape ("threshold-like")





Training an MLP

How to determine weights such that desired functition is performed?

Basic Idea: Compare (computed) output of MLP

$$\hat{\boldsymbol{y}} = \boldsymbol{f}^L\left(\boldsymbol{f}^{L-1}(\ldots \boldsymbol{f}^2\left(\boldsymbol{f}^1\left(\boldsymbol{x}\right)\right)\right)$$

to *desired* (ideal) output $\boldsymbol{y}$ and update weights such that $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$ become more similar.

Requires *loss function* that evaluates similarity of $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$:

- *Mean Square Error* (MSE):

$$\epsilon_{\text{MSE}} = \frac{1}{2} \cdot \sum_i (y_i - \hat{y}_i)^2$$

- *Cross-Entropy* (in comb. w. Softmax):

$$\epsilon_{\text{CE}} = -\sum_i y_i \log \hat{y}_i$$

Training an MLP II

How to update weights in order to reduce loss?

Compute gradient of loss wrt. the weights:

$$\frac{\partial \epsilon}{\partial w_{ij}^{l}}$$

Update weights in the negative direction of the gradient:

$$w_{ij}^{l} \leftarrow w_{ij}^{l} - \beta \cdot \frac{\partial \epsilon}{\partial w_{ij}^{l}}$$

$\Longrightarrow$ Training the network $\hat{=}$ Gradient Descent

Note: *Learning rate* β controls step size!

Stochastic Gradient Descent (SGD)

Gradient can be computed ...

(1) for complete training data (avg.)

(2) "online" for every new sample.

But *neither* works well!

⇒ Training takes forever!

⇒ Gradient is extremely noisy!

Solution:

Compute estimate of gradient for a small sub-set of the training data (so-called "mini batch")

Note: Samples are drawn *randomly* from the training set!

⇒ Gradient estimate is *stochastic*!

Stochastic Gradient Descent (SGD) II

Disadvantage of mini-batch processing:

Gradient is still somewhat noisy.

Improved Solution:

Introduce so-called *momentum* η $(0 \ll \eta < 1)$:

$$w_{ij}^{(t+1)} \leftarrow w_{ij}^{(t)} + \beta \left[(1-\eta) \Delta w_{ij}^{(t+1)} + \eta \Delta w_{ij}^{(t)} \right]$$

Note: Computes smoothed sequence of gradient estimates (i.e. sliding average with exponential decay).

Classifying Images with Neural Networks

Problem:

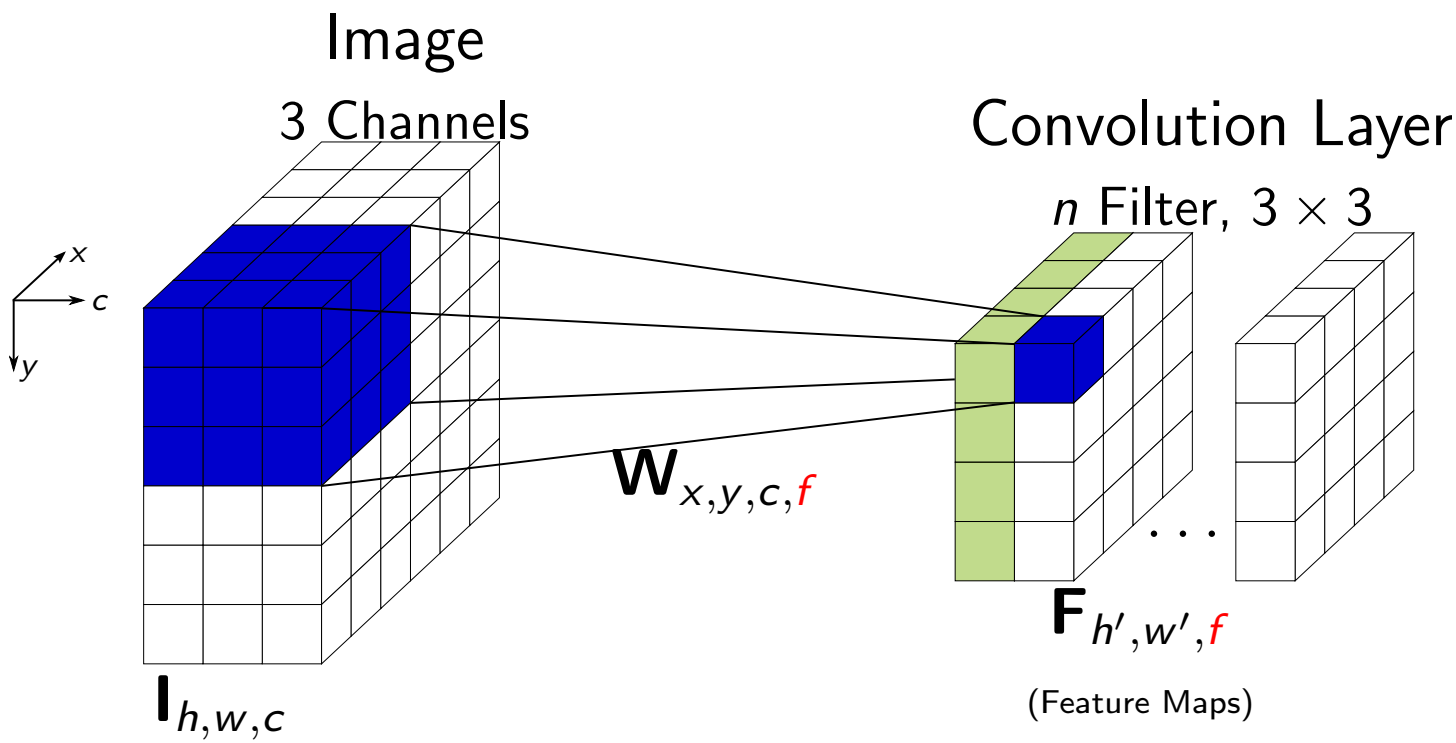
- Using MLPs for image classification is only possible for very small images (e.g. 28×28 pixels)
- Number of weights would explode for bigger images

Example: RGB Image of 224×224 pixels,
first hidden MLP layer has 768 neurons (small layer):
$224 \cdot 224 \cdot 3 \cdot 768 \approx 10^8$ weights in the first layer (441 MB)

Solution: Don't use fully connected layer but rather apply a small number of weights at all possible locations in the image

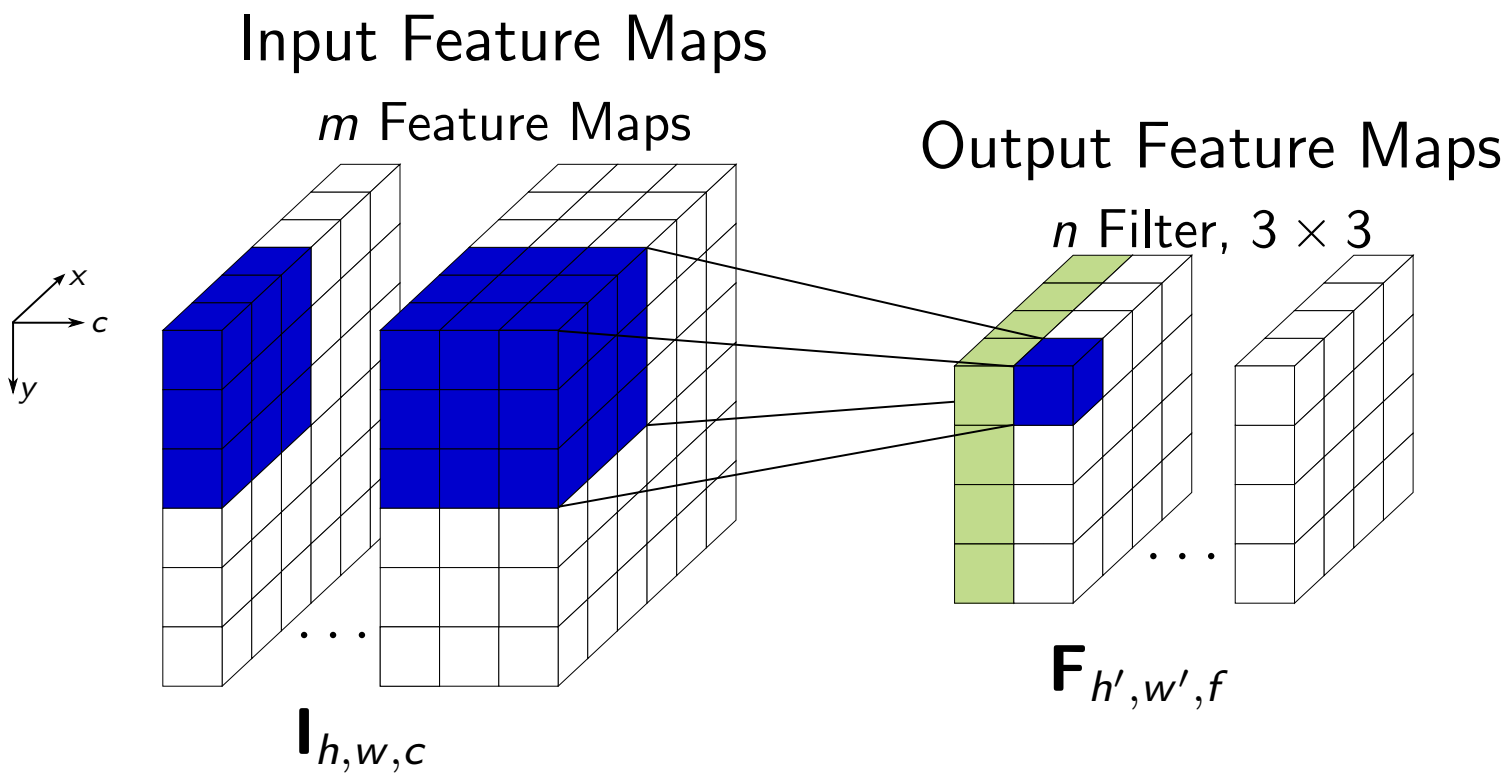
$\Rightarrow$ Convolutional Layer

Convolutional Layer

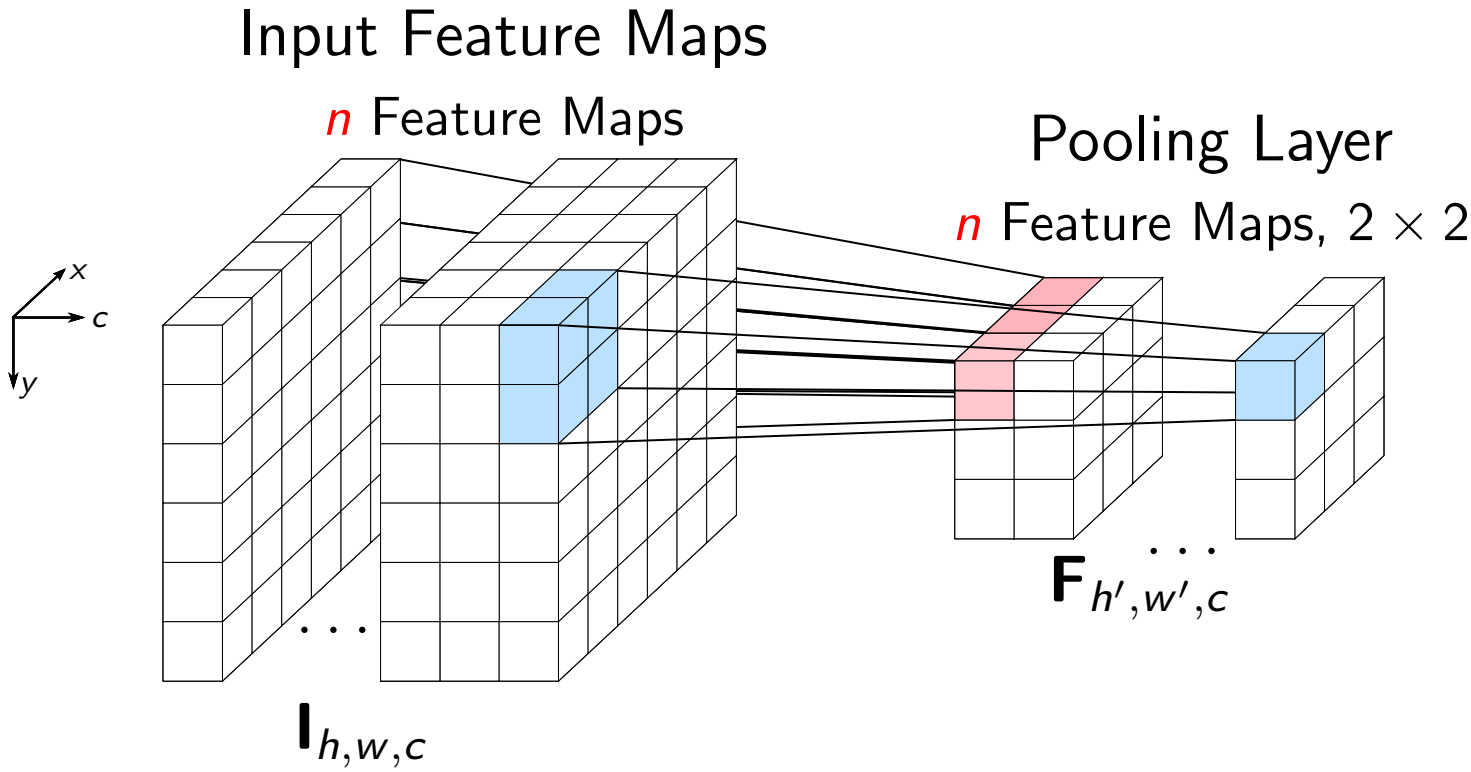


$$\mathbf{F}_{x,y,f} = \varphi\left(\sum_{c=1}^{K}\sum_{i=1}^{3}\sum_{j=1}^{3}\mathbf{W}_{i,j,c,f}\cdot\mathbf{I}_{x+i,y+j,c} + b_f\right)$$

Cascade of Convolutional Layers

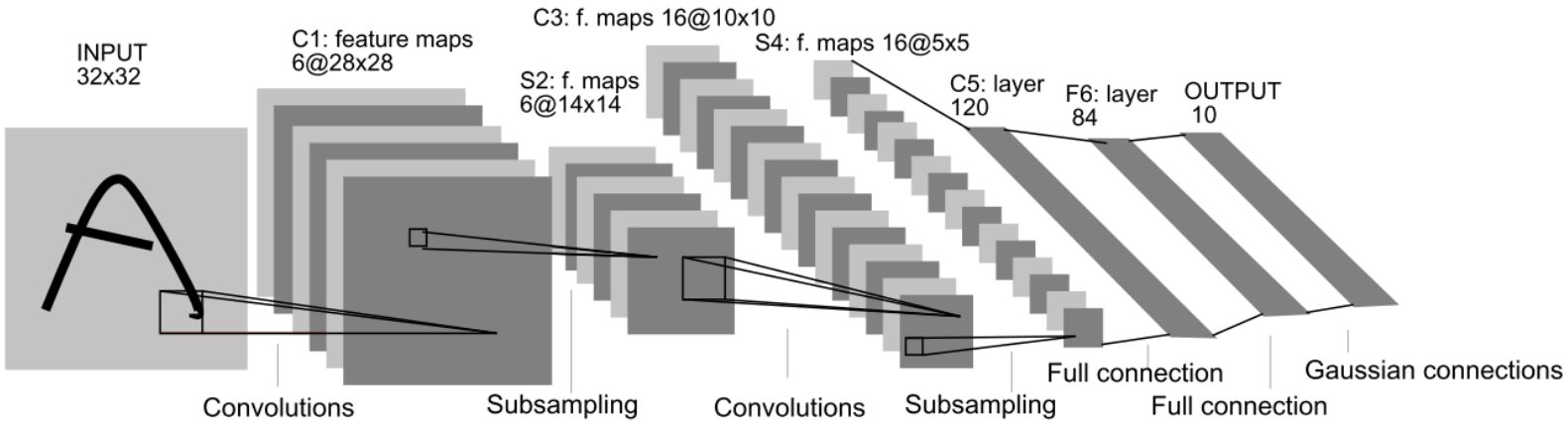


Pooling Layer



$$\mathbf{F}_{x,y,f} = \max_{i,j} \mathbf{I}_{x+i,y+j,f}$$

LeNet



(Source: [LeCun *et al.*, 1990])

- LeNet predicts one of 10 character classes for a given input image
- Subsampling = Pooling Layer
- Gaussian Connections = FC Layer + Euclidean Loss

Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L.D. Jackel: Handwritten Digit Recognition with a Back-Propagation Network, Neural Information Processing Systems, pp. 396–404, 1990.

Deep Learning

In general: Deeper network architectures perform better than shallower ones for vision tasks

Important: Only empirical evidence (no theoretical proofs)

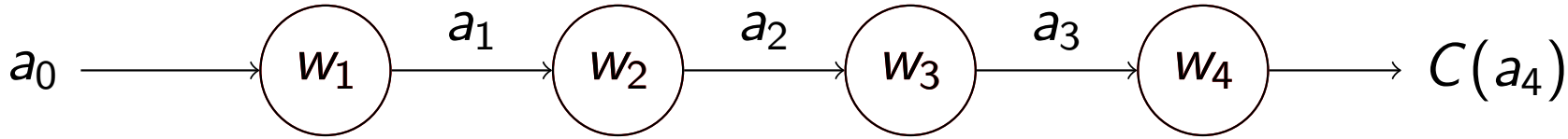
Technically: Deeper means more layers, not a deeper understanding

Even with high computation power and large datasets, Deep Learning did not really pick up until 2012!

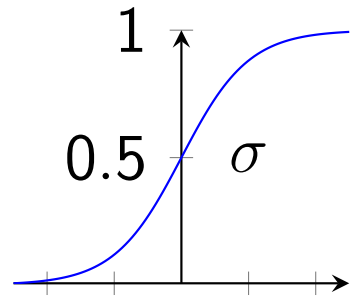
Why? Vanishing Gradient Problem

Vanishing Gradient Problem

Four neuron network, 1D input, 1D output



$$z_i = w_i a_{i-1} + b_i \qquad a_i = \sigma(z_i)$$

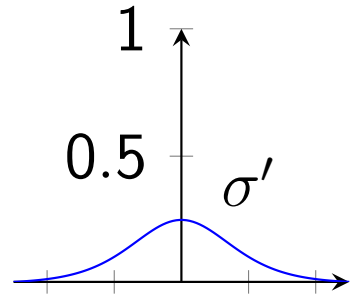


$$\frac{\partial C}{\partial w_1} = \frac{\partial C}{\partial z_4}\frac{\partial z_4}{\partial a_3} \cdot \frac{\partial a_3}{\partial z_3}\frac{\partial z_3}{\partial a_2} \cdot \frac{\partial a_2}{\partial z_2}\frac{\partial z_2}{\partial a_1} \cdot \frac{\partial a_1}{\partial z_1}\frac{\partial z_1}{\partial w_1}$$

$$= \sigma'(z_4) w_4 \cdot \sigma'(z_3) w_3 \cdot \sigma'(z_2) w_2 \cdot \sigma'(z_1) a_0$$

$$= \sigma'(z_4)\sigma'(z_3)\sigma'(z_2)\sigma'(z_1) \cdot w_4 w_3 w_2 a_0$$

$$= \underbrace{\sigma'(z_4)\sigma'(z_3)\sigma'(z_2)\sigma'(z_1)}_{\leq \frac{1}{4}\cdot\frac{1}{4}\cdot\frac{1}{4}\cdot\frac{1}{4}} \cdot w_4 w_3 w_2 a_0$$



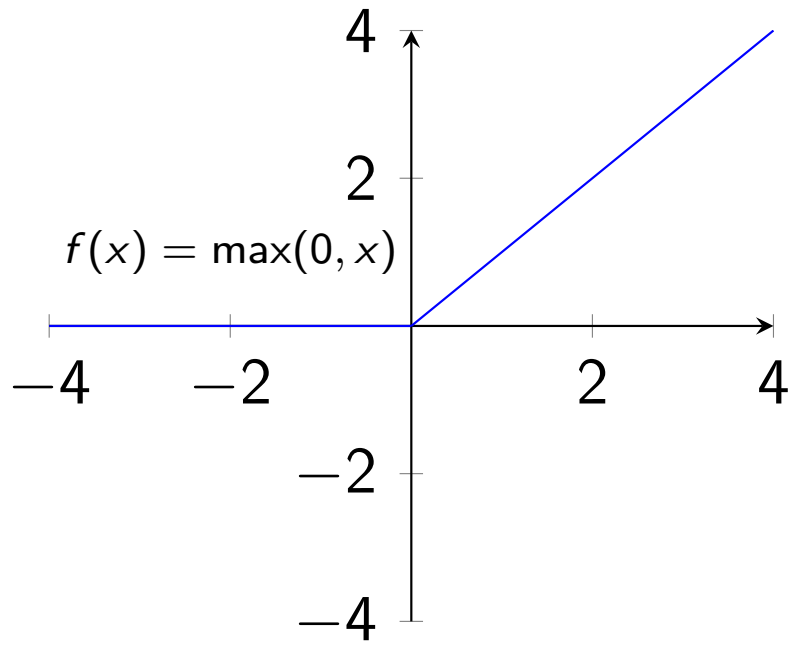
$$\leq \frac{1}{4^4} \cdot w_4 w_3 w_2 a_0$$

Vanishing Gradient Problem

- Derivative of sigmoidal activation functions < 1
- Exponential decay of gradient magnitude

Desirable: Activation function with derivative $= 1$ but non-linear ($> 1 =$ exploding gradient)

Solution: Rectified Linear Unit (ReLU) [Glorot & Bengio 2010]



How to Get Along With Limited Training Data?

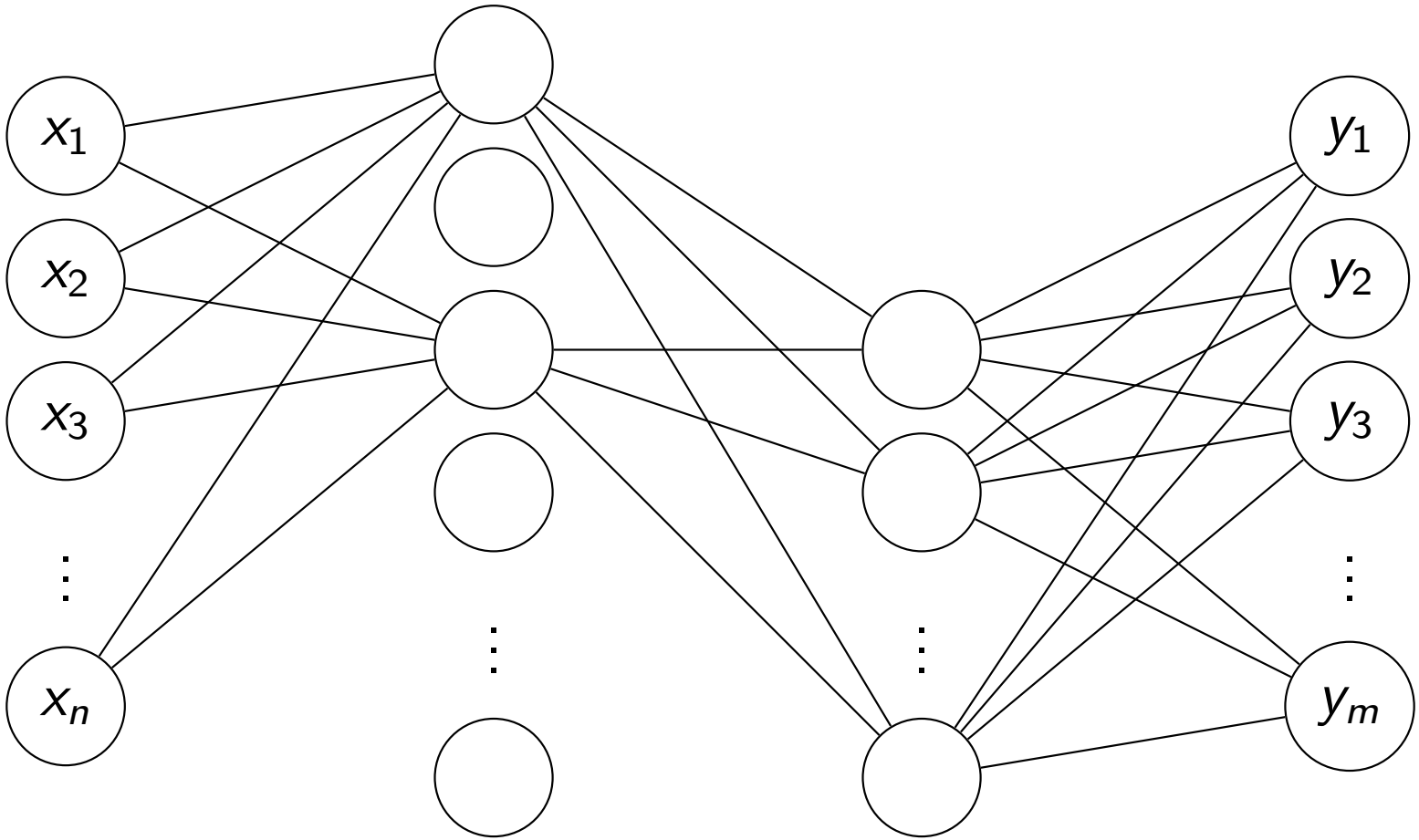
Problem: CNNs easily contain billions of parameters (weights)!
$\Rightarrow$ Could easily learn training samples "by heart".

Solution: Apply *Regularization* during training

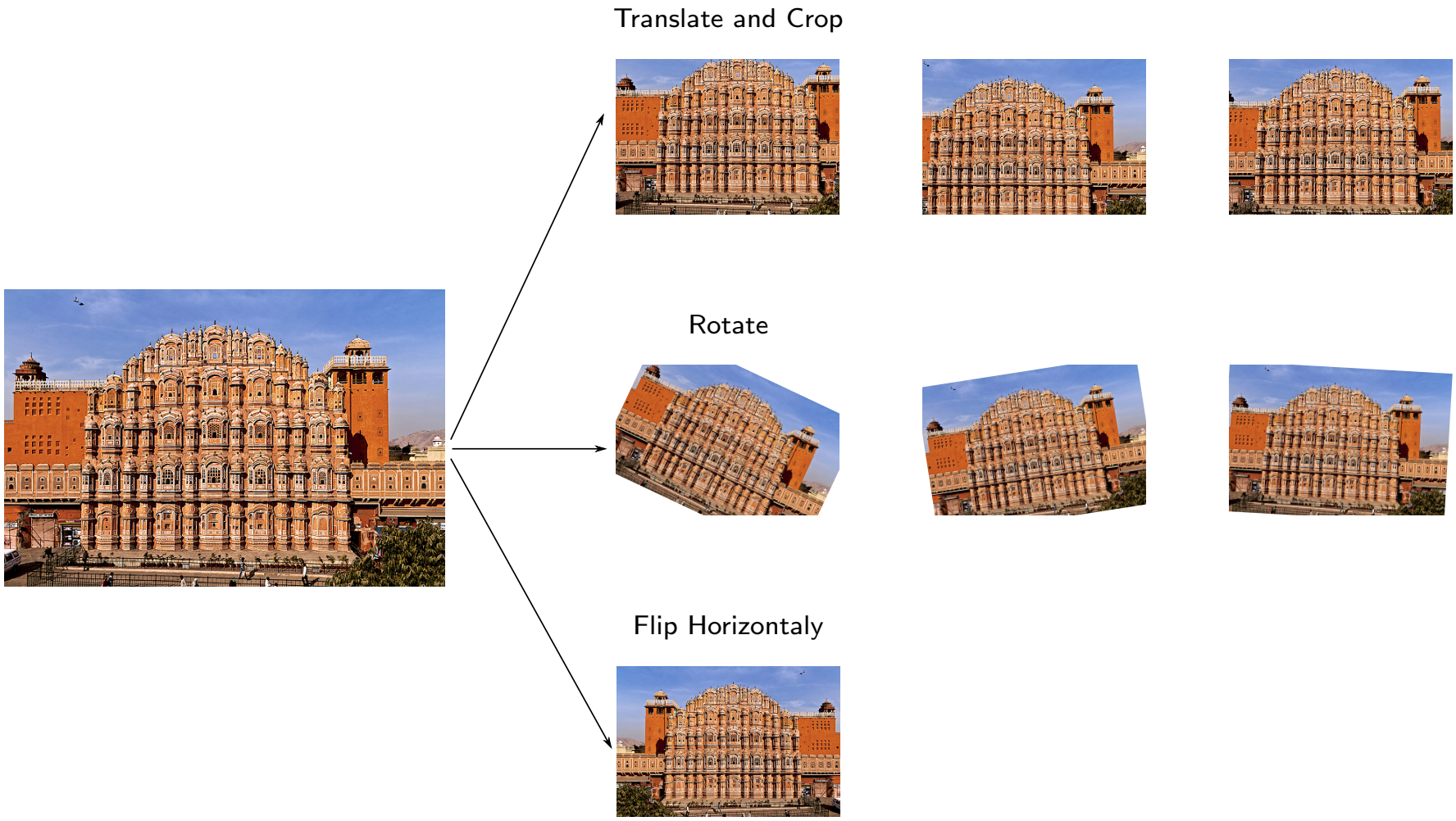
Fundamental Techniques:

- *Convolutional layers*
- *Dropout*
  Randomly set outputs of neurons to zero
  (usually 50% of fully connected layers)
- *Data Agumentation:*
  Generate new, slightly different training samples from existing ones by certain transforms
  (e.g. slight translations, rotations, ...)

Dropout

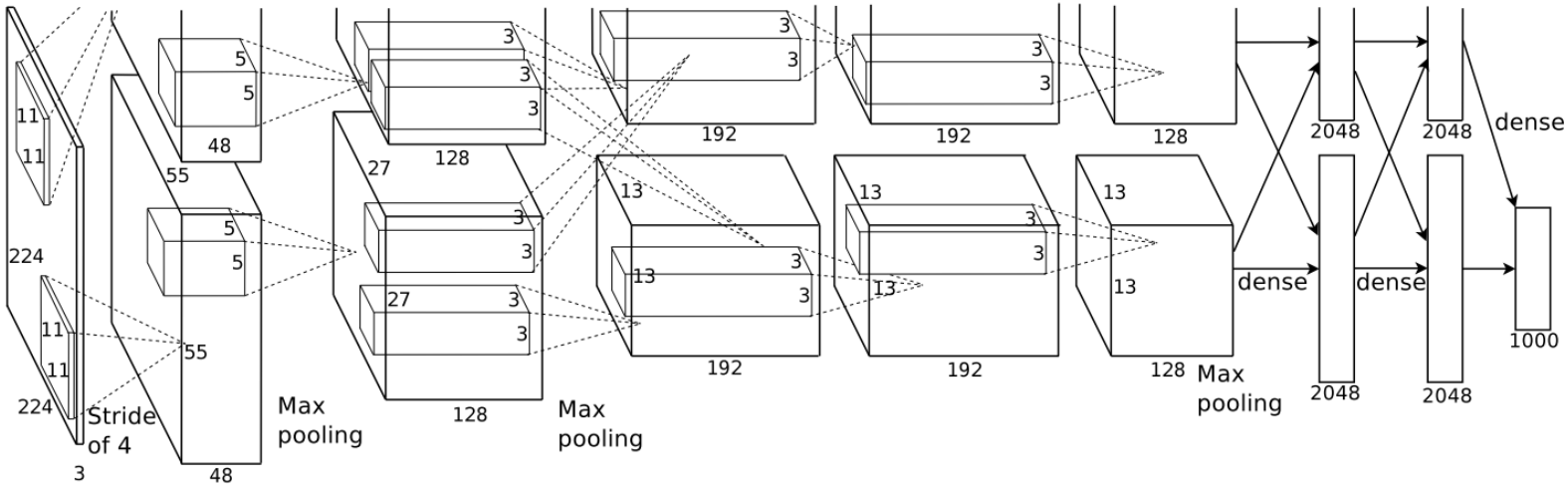


Data Augmentation



Note: Usually different augmentation techniques are mixed to create a single augmented image

Well-known Deep Learning Architectures: AlexNet

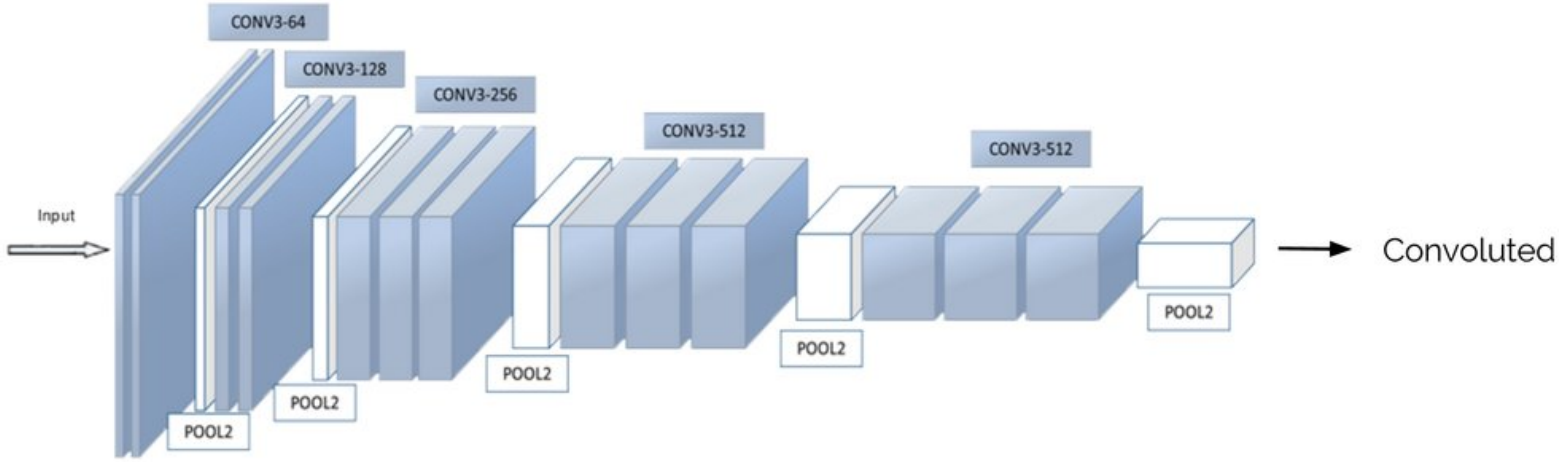


(Source: [Krizhevsky *et al.*, 2012])

- CNN which kicked off the current Deep Learning hype
- Architecture similar to LeNet but more layers/parameters
- Trained on two graphic cards for over a week on ImageNet

A. Krizhevsky, I. Sutskever, G. E. Hinton: ImageNet Classification with Deep Convolutional Neural Networks, Neural Information Processing Systems, pp. 1097–1105, 2012.

Well-known Deep Learning Architectures: VGGNet



(Source: http://html.scirp.org/)

- First CNN to use only 3×3 convolutions (standard for current CNNs)
- Low number of filters in the early layers, high number of filters in the later layers
- Anytime pooling is applied, the number of filters is doubled

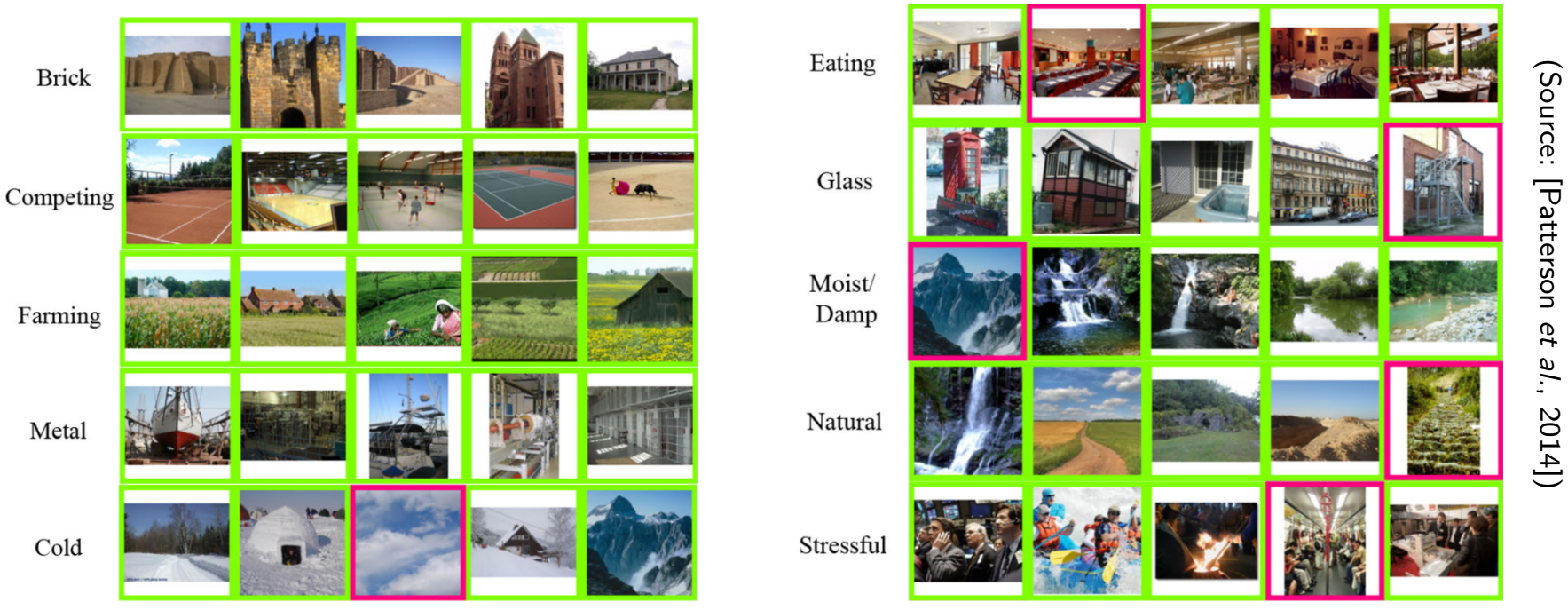
K. Simonyan, A. Zisserman: Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv, 2014.

Applications

Image Retrieval

Retrieval of Attributes

Goal: Retrieve images based on a given attribute

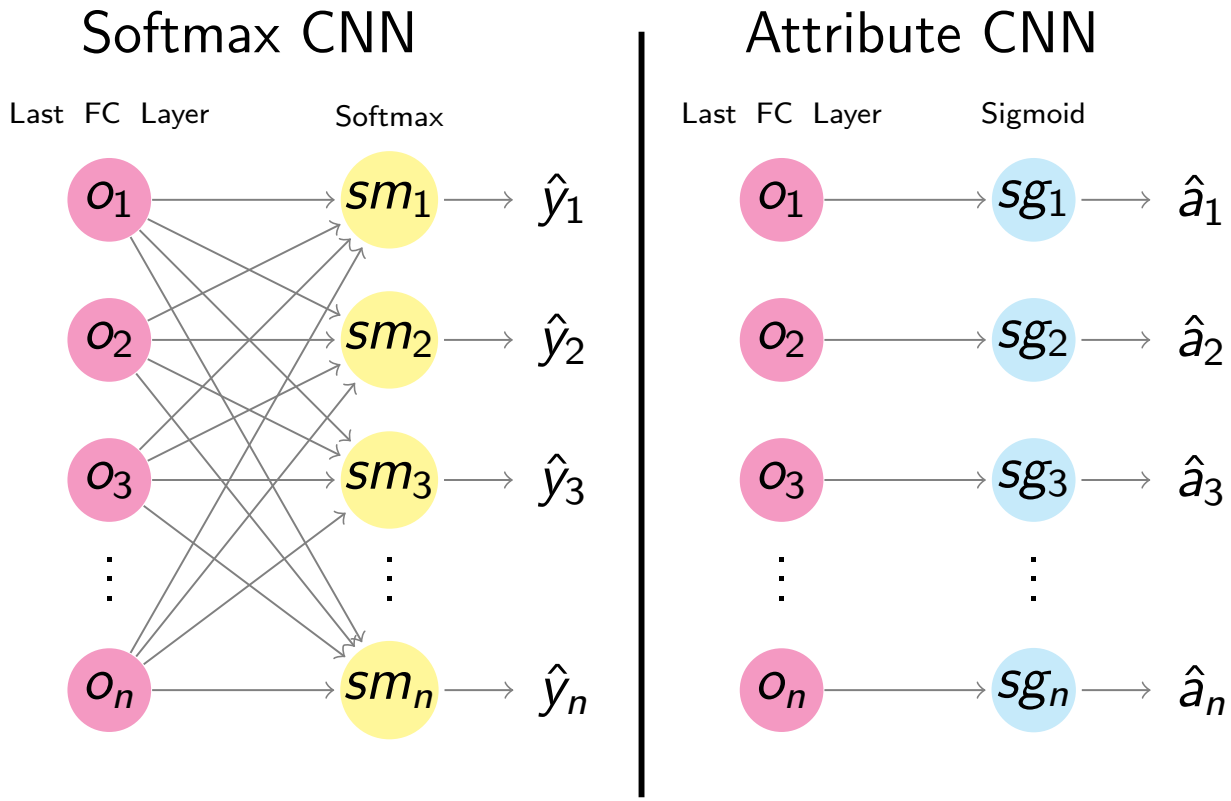


(Source: [Patterson *et al.*, 2014])

G. Patterson, C. Xu, H. Su, J. Hays: The SUN Attribute Database: Beyond Categories for Deeper Scene Understanding Int. Journal of Computer Vision, 108(1-2):59–81, 2014.

Predicting Attribute Representations

- In order to classify attributes, replace softmax with a sigmoid activation
- Each output neuron predicts one attribute



Evaluation Measures

Accuracy: Correct classification of an attribute

Mean Average Precision: Measure for assessing the relevance and order of a retrieval list

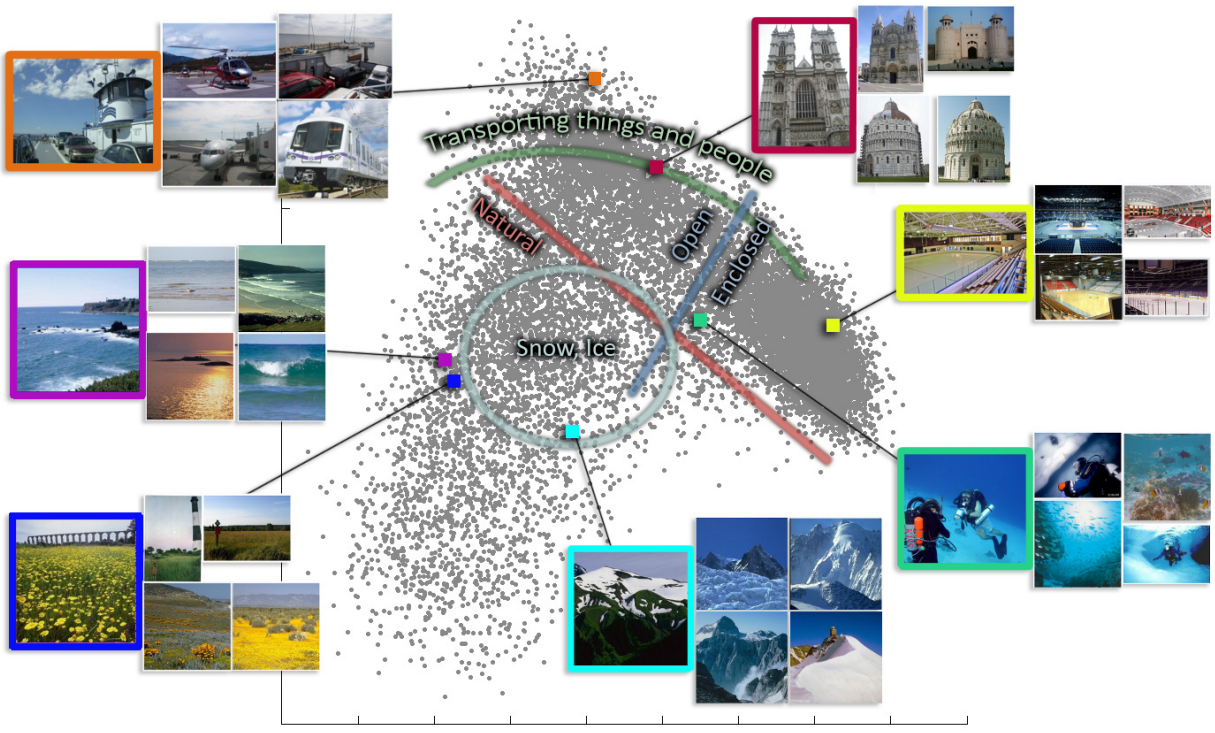
Computing mAP:

- Average Precision: Precision averaged at different recall levels (cf. e.g. [3, pp 140-141]):

$$\frac{\sum_{k=1}^{n} \text{Prec}_k \times \text{rel}(k)}{\#\text{Relevant Items in Dataset}}$$

  $\text{rel}(k)$: Relvance of item k, Prec_k: Precision at cut-off k
- Mean Average Precision (mAP): Average Precision averaged over all queries considered

Scene Attributes — SUN Database



(Source: [Patterson *et al.*, 2014])

102 attributes:

- driving
- wood
- using tools
- asphalt
- pavement
- sand
- warm
- cluttered space
- ...

G. Patterson, C. Xu, H. Su, J. Hays: The SUN Attribute Database: Beyond Categories for Deeper Scene Understanding
Int. Journal of Computer Vision, 108(1-2):59–81, 2014.

Experimental Evaluation

Attribute-based retrieval performance on SUN in mAP [%]

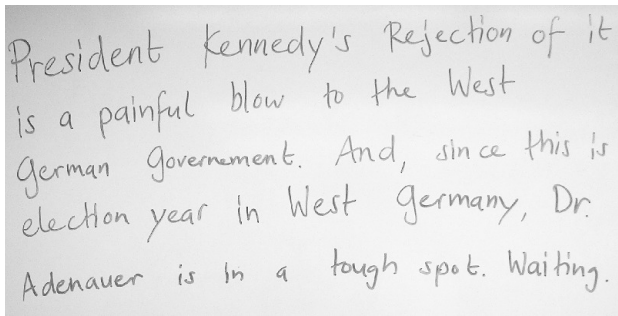
Network	Affordances	Materials	Surfaces	Spatial Envelope	Total
SVMs [15]	44	51	50	62	50
VGGNet	54.2	**58.6**	58.5	65.2	58.0
VGGNet + SPP	**55.7**	58.4	**58.7**	**65.7**	58.6

R. Grzeszick, S. Sudholt, G. A. Fink: *Optimistic and Pessimistic Neural Networks for Scene and Object Recognition*, CoRR, arXiv:1609.07982v2, 2016.

Word Spotting

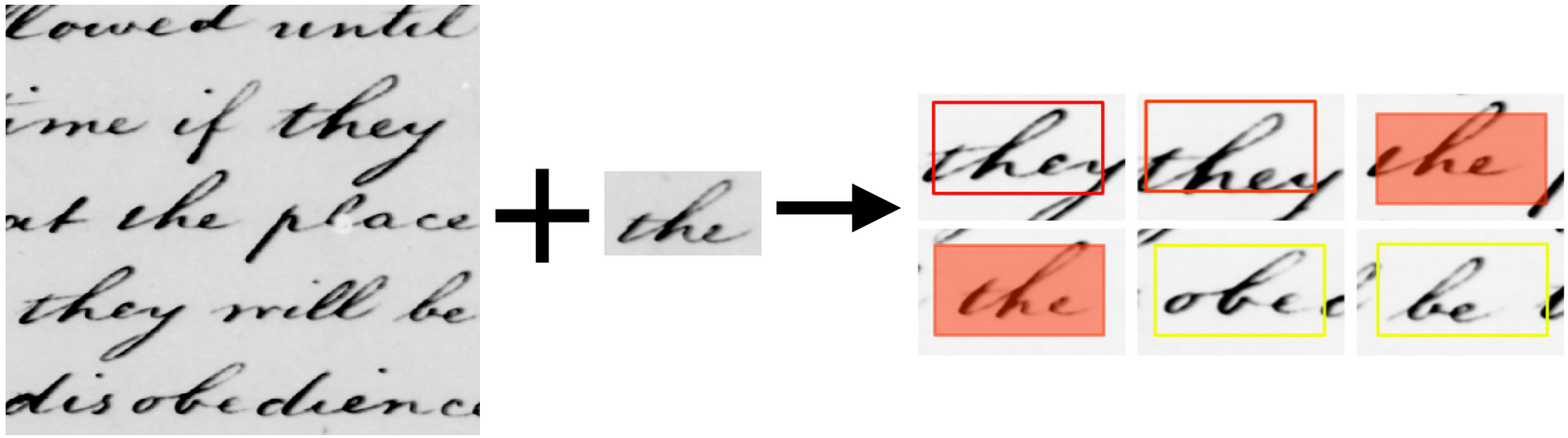
Introduction: Transcription vs. Retrieval

Document Transcription (= "classical" recognition)



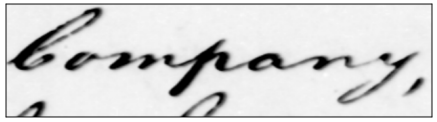
President Kennedy is Rejection of it is a painful blow to the West German Government And since this is selection year in West Germany of Adenauer is in a tough spot waiting

Document Retrieval (aka "Word Spotting")



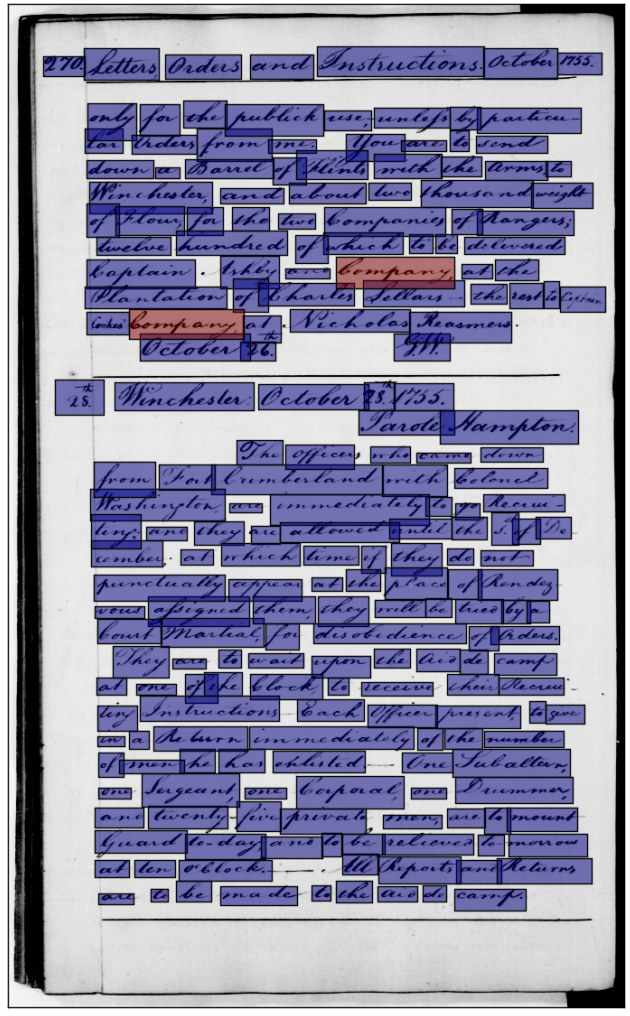
Word Spotting: Tasks

Query by Example



Query by String

Company

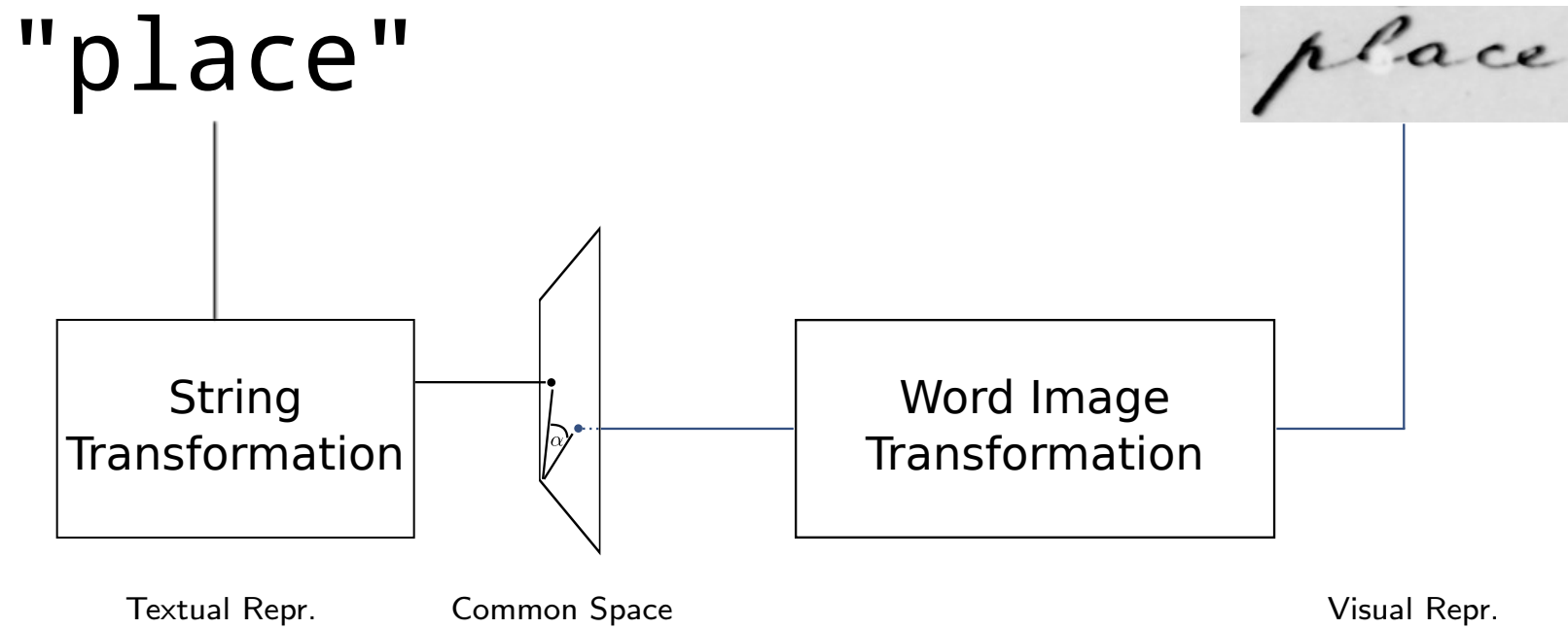


Images from *The George Washington Papers at the Library of Congress, 1741-1799*

Subspace Representations for Word Spotting

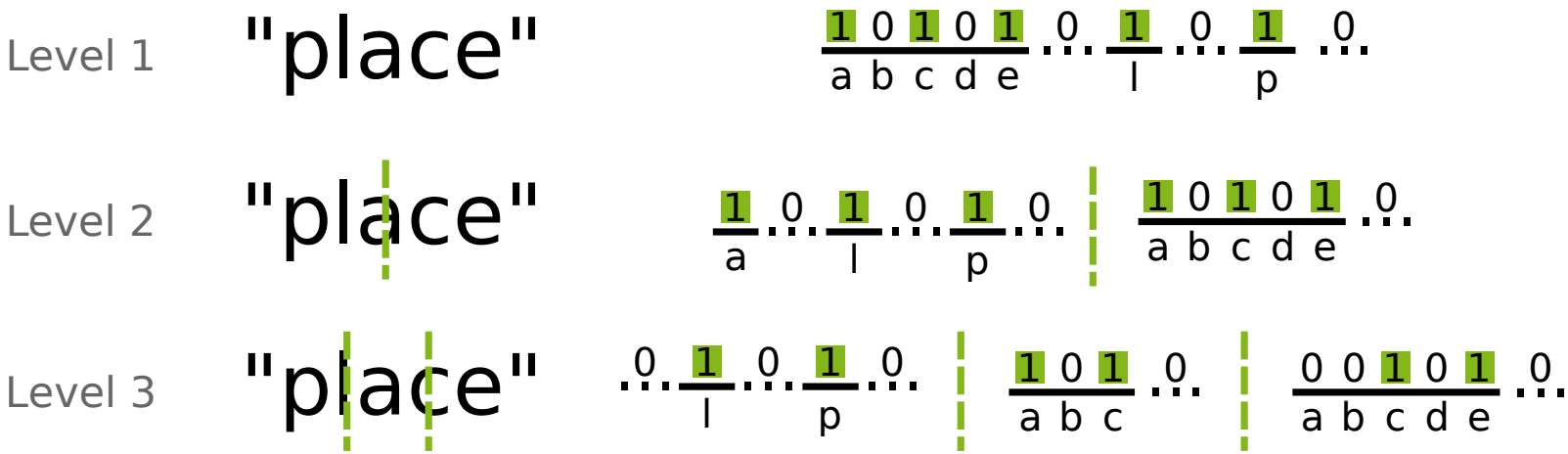
Idea: Project both textual and visual representation into a *common* space

Benefits: QbE and QbS are now a simple nearest neighbor search



J. Almazán, A. Gordo, A. Fornés and E. Valveny: *Word Spotting and Recognition with Embedded Attributes*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 36, no. 12, pp. 2552-2566, 2014.

Pyramidal Histogram of Characters (PHOC)



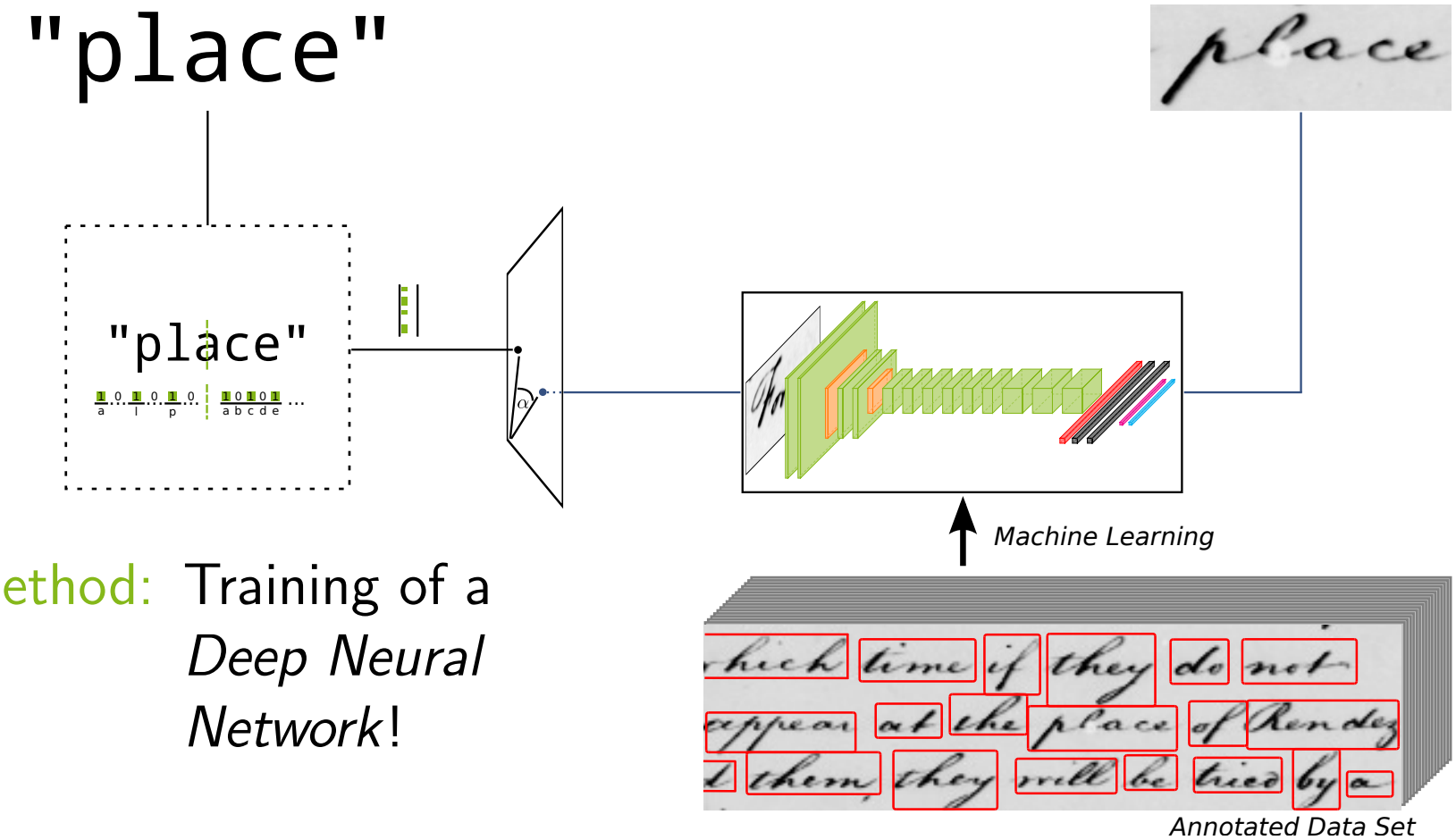
- Concatenate histograms for all levels to form PHOC
- Levels used by Almazán *et al.*: 2,3,4 and 5
- 26 Characters + 10 Digits
- PHOC $\in \{0,1\}^{604}$

J. Almazán, A. Gordo, A. Fornés and E. Valveny: *Word Spotting and Recognition with Embedded Attributes*, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 36, no. 12, pp. 2552-2566, 2014.

Wordspotting: Modeling

Idea: Learning the word-image transform from examples!

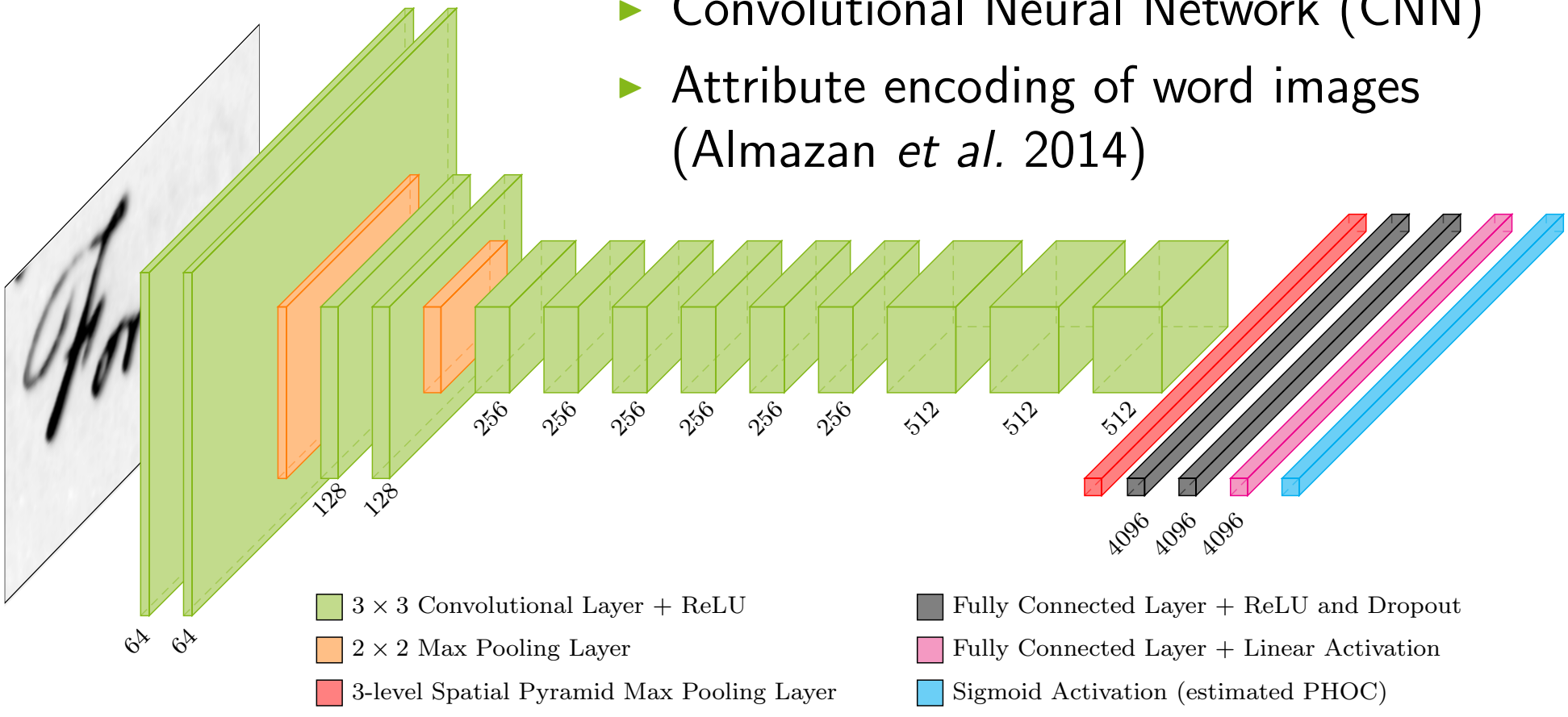
⇒ *Annotated* data set required!



Method: Training of a *Deep Neural Network*!

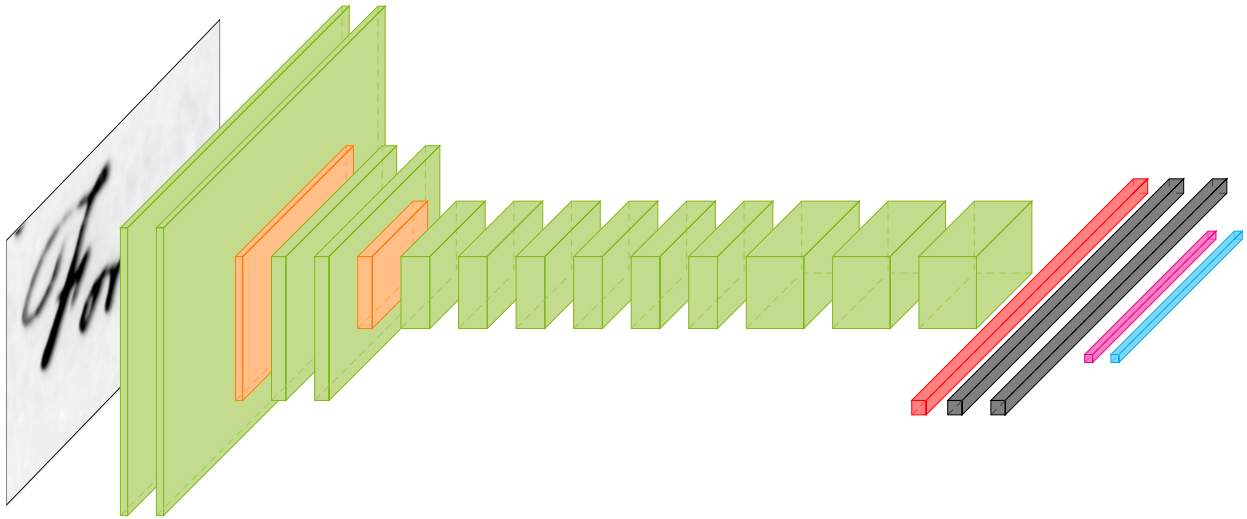
Wordspotting: Deep Neural Networks

- Convolutional Neural Network (CNN)
- Attribute encoding of word images (Almazan *et al.* 2014)



(Sudholt & Fink: ICFHR, 2016, Winner of the *Best Paper Award*!)

Word Spotting: The PHOCNet



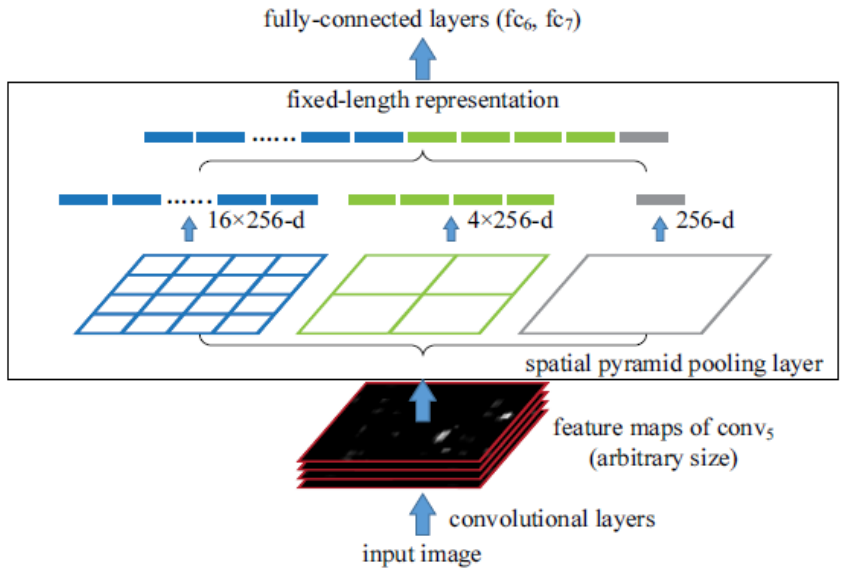
- Predicts PHOC representation for given word image
- Only uses 3×3 convolutions (regularization)
- Can accept (almost) arbitrarily sized images (no rescaling or cropping)

S. Sudholt, G. A. Fink: PHOCNet: A Deep Convolutional Neural Network for Word Spotting in Handwritten Documents, Proc. Int. Conf. on Frontiers in Handwriting Recognition, Shenzhen, China, 2016.

Handling Varying Input-Image Sizes

- Convolutional layers can deal with arbitrary image sizes
- Only MLP part has a problem with changing image sizes

Solution: Apply Spatial Pyramid concept to last convolutional output



(Source: [He *et al.*, 2014])

K. He, X. Zhang, S. Ren, J. Sun: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, Proc. European Conference on Computer Vision, pp. 346–361, 2014.

Experimental Evaluation

Segmentation-based Word Spotting Performance in mAP [%]

Method	GW		IAM		Esposalles		IFN/ENIT	
	QbE	QbS	QbE	QbS	QbE	QbS	QbE	QbS
PHOCNet	97.96	**97.92**	**85.50**	**93.42**	**97.40**	**94.89**	**96.66**	**94.92**
Deep Feat. Emb. [10]	94.41	92.84	84.24	91.58	-	-	-	-
Triplet-CNN [21]	**98.00**	93.69	81.58	89.49	-	-	-	-
Attribute SVM [2]	93.04	91.29	55.73	73.72	-	-	-	-
LSA Embedding [1]	-	56.54	-	-	-	-	-	-
Finetuned CNN [18]	-	-	46.53	-	-	-	-	-
Softmax CNN	78.24	-	48.67	-	89.38	-	91.78	-
BLSTM [5]	-	84.00	-	78.00	-	-	-	-
SC-HMM [16]	53.10	-	-	-	-	-	41.60	-

Human Activity Recognition

Introduction

What is Human Activity Recognition (HAR)?

Goal: Classification of human movements from video or measurements of specialized sensors (e.g. IMUs)

Component of smart assistive technologies in ...

- Smart homes
- Health support
- Rehabilitation
- Industry

Our application:

Analzying the process of Order Picking

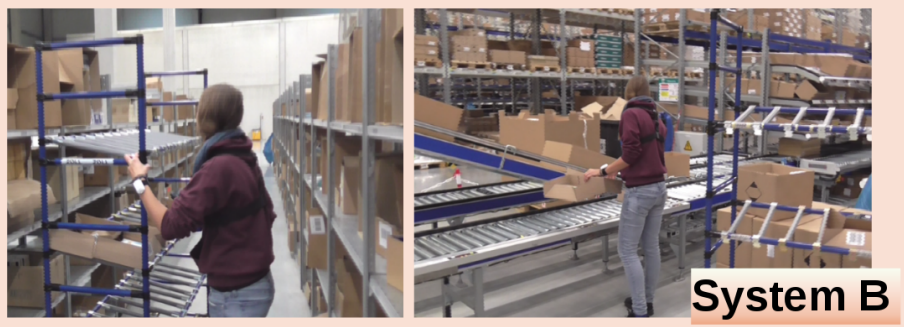
... i.e. letting someone go shopping for you

Order Picking: Settings and Data

Settings dependent on warehouse and process organization!

- Paper list
- Cart
- Boxes

- Mobile Terminal
- Cart
- Cartons

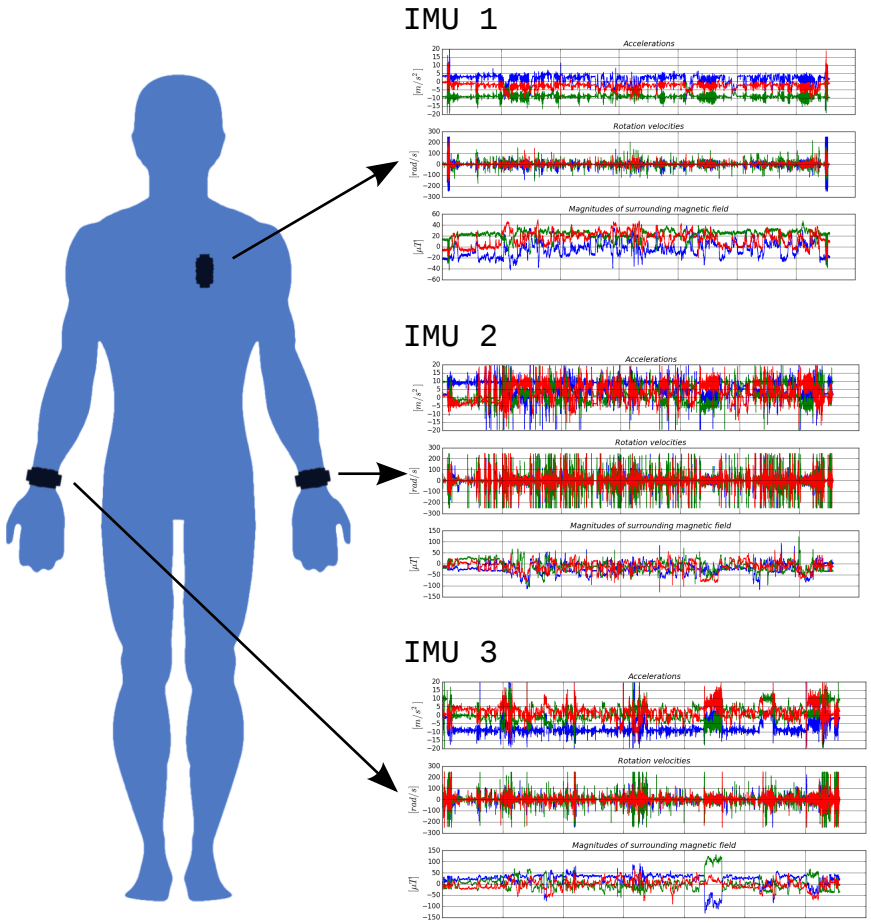


(Source: [Feldhorst *et al.*, 2016])

S. Feldhorst, M. Masoudinejad, M. ten Hompel, G. A. Fink: *Motion Classification for Analyzing the Order Picking Process Using Mobile Sensors*, Proc. Int. Conf. Pattern Recognition Applications and Methods (ICPRAM), Rome, 2016.

Order Picking: Settings and Data

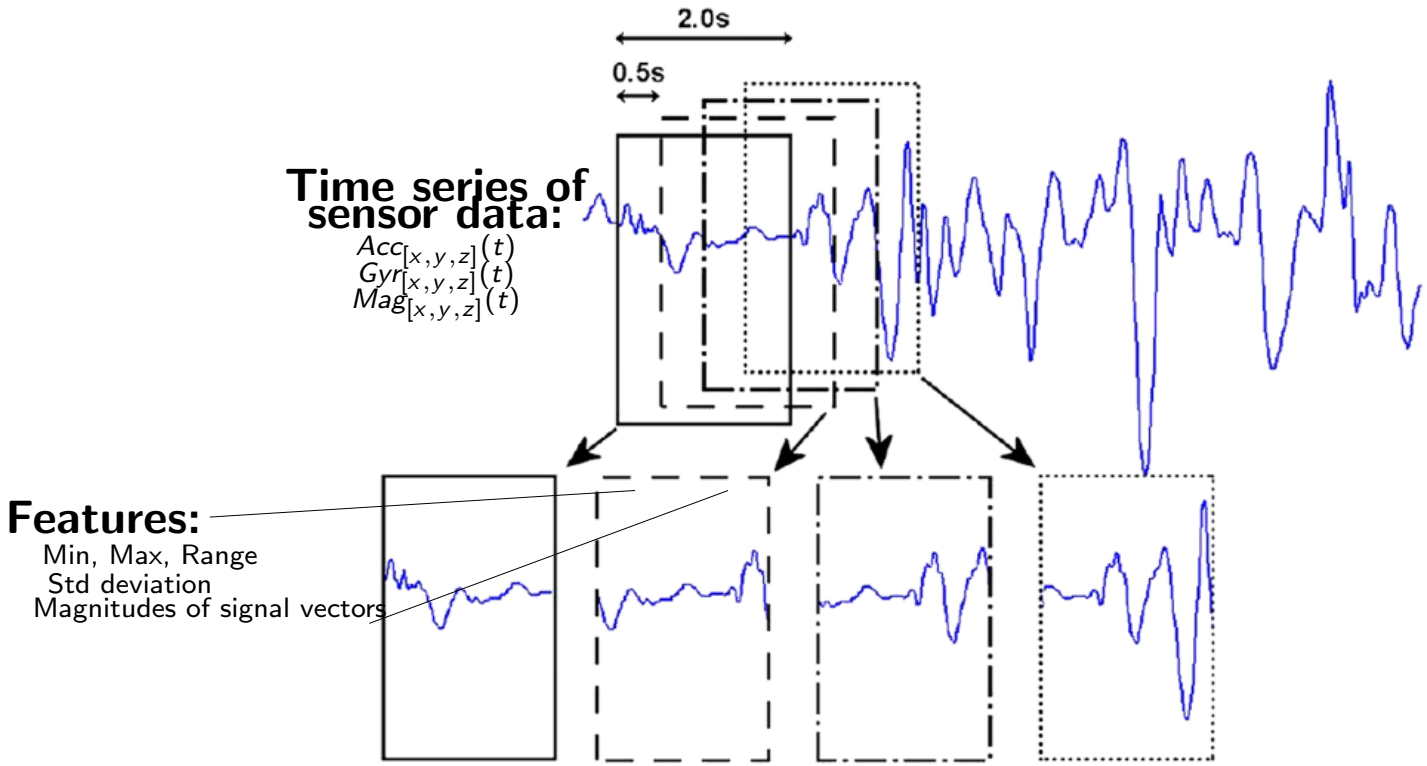
Data provided by Inertial Measurements Units (IMUs)



IMU = accelerometer + gyroscope + magnetometer (each providing readings in x/y/z)

Order Picking: Activity Recognition

Traditional approach: Sliding window, features, classification



(Source: [Feldhorst *et al.*, 2014])

S. Feldhorst, M. Masoudinejad, M. ten Hompel, G. A. Fink: *Motion Classification for Analyzing the Order Picking Process Using Mobile Sensors*, Proc. Int. Conf. Pattern Recognition Applications and Methods (ICPRAM), Rome, 2016.

Learning Attribute Representations

Idea: Iteratively improve existing attribute representation

Method: Evolutionary Algorithm (EA)

(attribute prepresentation = gene)

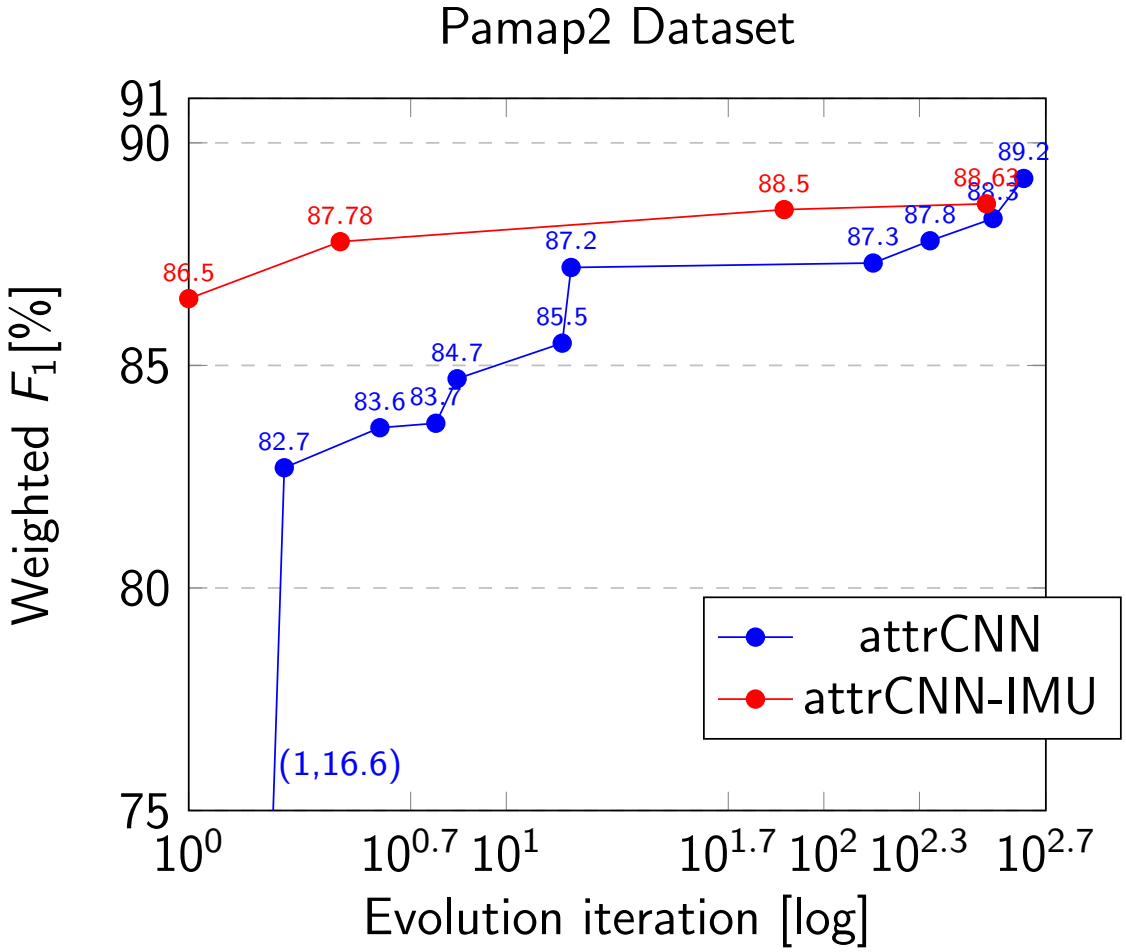
Starting Point: Random attribute representation

Fitness evaluated by training and validating a deep network

Evaluation measure: Weighted F_1

$$F_1 = \sum_i 2 \times \frac{n_i}{N} \times \frac{precision_i \times recall_i}{precision_i + recall_i}$$

Learning Attribute Representations II



Weighted F_1[%] of the attribute representations during optimization

Activity Recognition: First Results

Performance comparison of CNNs with learned attribute representations and state-of-the-art networks (weighted F_1[%])

Architecture	Pamap2	Locomotion	Gestures
CNN [Ordóñez *et al.*]	87.37	87.8	85.1
CNN [Hammerla *et al.*]	87.2	-	90.8
CNN-IMU [Grzeszick *et al.*]	89.01	88.23	92.15
attrCNN random	84.72	85.9	88.96
attrCNN-IMU random	86.26	86.85	89.92
attrCNN evol	90.55	**90.0**	91.94
attrCNN-IMU evol	**90.88**	89.75	**92.9**

F. Moya Rueda, G. A. Fink: *Learning Attribute Representations for Human Activity Recognition*, Proc. Int. Conf. Pattern Recognition, 2018, submitted.

Summary

Summary

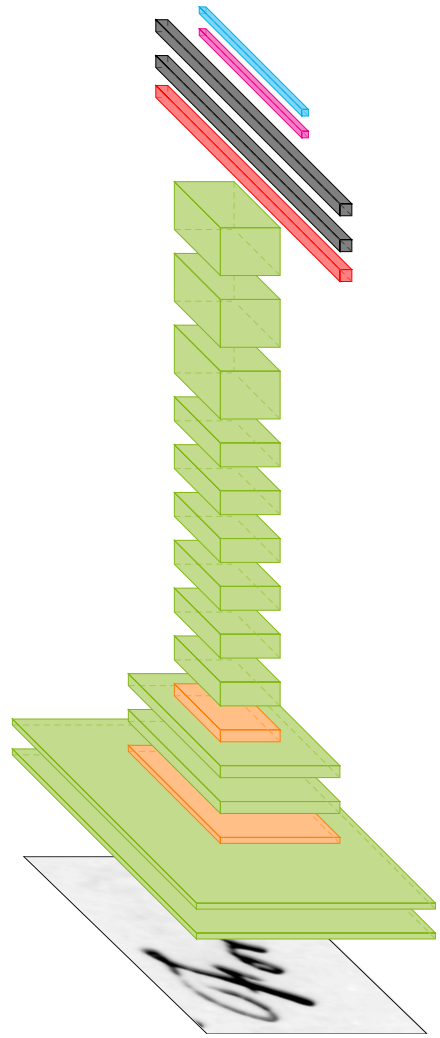
Attributes are beneficial for pattern recognition:

- Provide a finite-dimensional description of a potentially unlimited set of categories
- Can be shared between categories

Convolutional Neural Networks (CNNs) ...

- Can successfully be used to predict attribute representations from sensor data
- Can exploit the sharing of attributes for improved performance

Attribute Representations can even be learned!



References I

David Aldavert, Marcal Rusinol, Ricardo Toledo, and Josep Llados.
Integrating visual and textual cues for query-by-string word spotting.
In *International Conference on Document Analysis and Recognition*, pages 511–515, 2013.

Jon Almazán, Albert Gordo, Alicia Fornés, and Ernest Valveny.
Word spotting and recognition with embedded attributes.
IEEE Trans. on Pattern Analysis and Machine Intelligence, 36(12):2552–2566, 2014.

References II


Ricardo A. Baeza-Yates and Berthier Ribeiro-Neto.
Modern Information Retrieval.
Pearson Education Limited, Harlow, England, 2 edition, 2011.


A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth.
Describing objects by their attributes.
In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785, 2009.

References III

Volkmar Frinken, Andreas Fischer, R. Manmatha, and Horst Bunke.
A novel word spotting method based on recurrent neural networks.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 34:211–224, 2012.

Xavier Glorot and Yoshua Bengio.
Understanding the difficulty of training deep feedforward neural networks.
AISTATS, 9:249–256, 2010.

References IV


Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.
Spatial pyramid pooling in deep convolutional networks for visual recognition.
European Conference on Computer Vision, pages 346–361, 2014.


Kurt Hornik, Maxwell Stinchcombe, and Halbert White.
Multilayer feedforward networks are universal approximators.
Neural Networks, 2(5):359–366, 1989.

References V

Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell.
Caffe: Convolutional architecture for fast feature embedding.
arXiv preprint arXiv:1408.5093, 2014.

Praveen Krishnan, Kartik Dutta, and C.V. Jawahar.
Deep feature embedding for accurate recognition and retrieval of handwritten text.
In *International Conference on Frontiers in Handwriting Recognition*, pages 289–294, 2016.

References VI


Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton.
Imagenet classification with deep convolutional neural networks.
In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Neural Information Processing Systems*, pages 1097–1105, 2012.


Christoph H. Lampert, Hannes Nickisch, and Stefan Harmeling.
Learning to detect unseen object classes by between-class attribute transfer.
In *Proc. of the IEEE Comp. Soc. Conf. on Computer Vision and Pattern Recognition*, pages 951–958, 2009.

References VII


Yann LeCun, Yoshua Bengio, and Geoffrey Hinton.
Deep learning.
Nature, 521:436–444, 2015.


Yann LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel.
Handwritten digit recognition with a back-propagation network.
Neural Information Processing Systems, pages 396–404, 1990.

References VIII


Genevieve Patterson, Chen Xu, Hang Su, and James Hays.
The SUN Attribute Database: Beyond Categories for Deeper Scene Understanding.
International Journal of Computer Vision, 108(1-2):59–81, 2014.


José A. Rodríguez-Serrano and Florent Perronnin.
A model-based sequence similarity with application to handwritten word spotting.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(11):2108–2120, 2012.

References IX


F. Rosenblatt.
The perceptron: A probabilistic model for information storage and organization in the brain.
Psychological Review, 65(6):386–408, 1958.


Arjun Sharma and Sankar K. Pramod.
Adapting off-the-shelf CNNs for word spotting & recognition.
In *International Conference on Document Image Analysis*, pages 986–990, 2015.


Karen Simonyan and Andrew Zisserman.
Very deep convolutional networks for large-scale image recognition.
arXiv, pages 1–13, 2014.

References X


Sebastian Sudholt and Gernot A. Fink.
PHOCNet: A deep convolutional neural network for word spotting in handwritten documents.
In *Proc. Int. Conf. on Frontiers in Handwriting Recognition*, Shenzhen, China, 2016.


Tomas Wilkinson and Anders Brun.
Semantic and verbatim word spotting using deep neural networks.
In *International Conference on Frontiers in Handwriting Recognition*, pages 307–312, 2016.