10707
Deep Learning

Russ Salakhutdinov

Machine Learning Department
rsalakhu@cs.cmu.edu
http://www.cs.cmu.edu/~rsalakhu/10707/

Convolutional Networks II

Used Resources

- **Disclaimer**: Much of the material in this lecture was borrowed from Hugo Larochelle's class on Neural Networks:
https://sites.google.com/site/deeplearningsummerschool2016/

- Some tutorial slides were borrowed from Rob Fergus' CIFAR tutorial on ConvNets:
https://sites.google.com/site/deeplearningsummerschool2016/speakers

- Some slides were borrowed from Marc'Aurelio Ranzato's CVPR 2014 tutorial on Convolutional Nets
https://sites.google.com/site/lsvrtutorialcvpr14/home/deeplearning

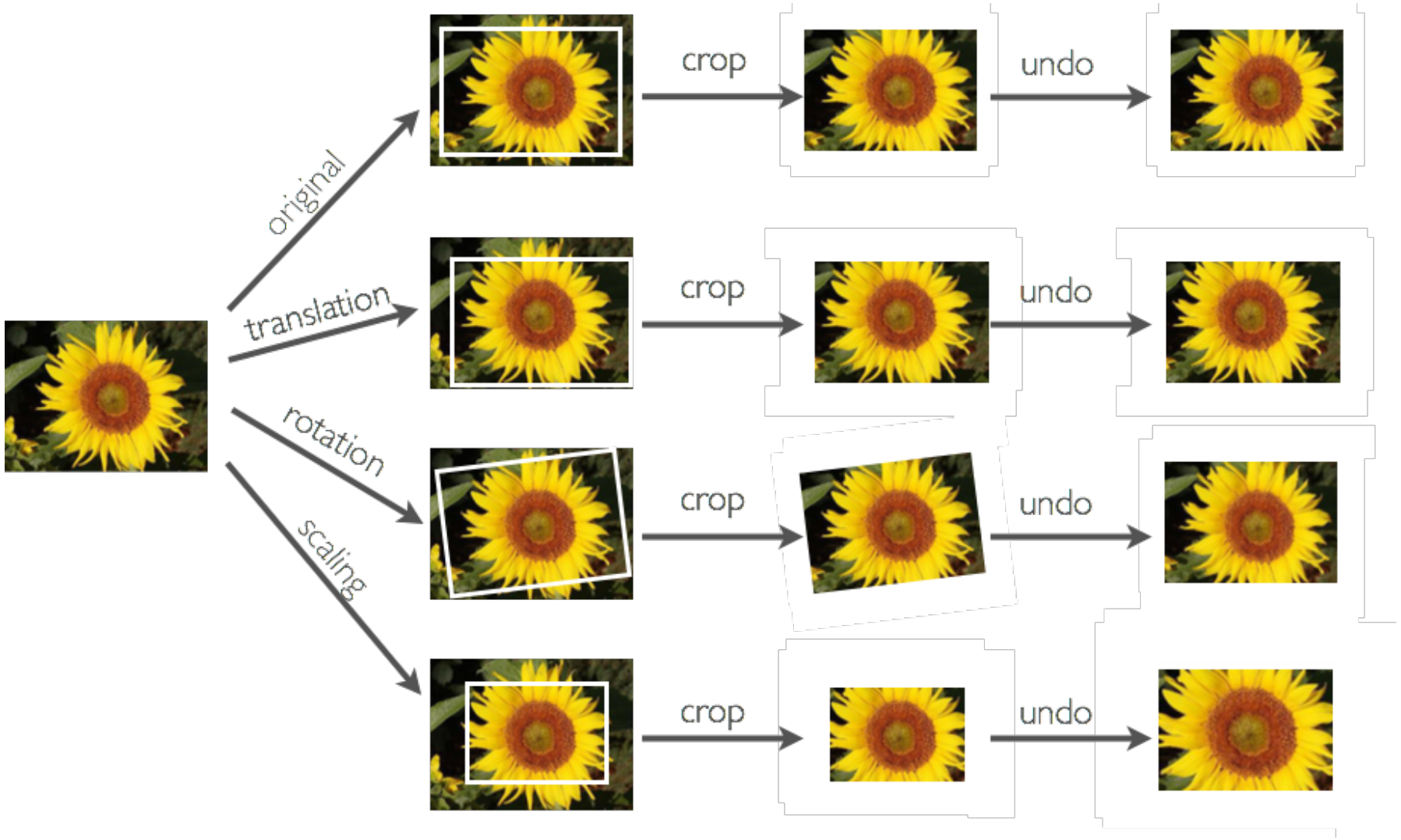
Invariance by Dataset Expansion

- Invariances built-in in convolutional network:
  - small translations: due to convolution and max pooling
  - small illumination changes: due to local contrast normalization

- It is not invariant to other important variations such as rotations and scale changes

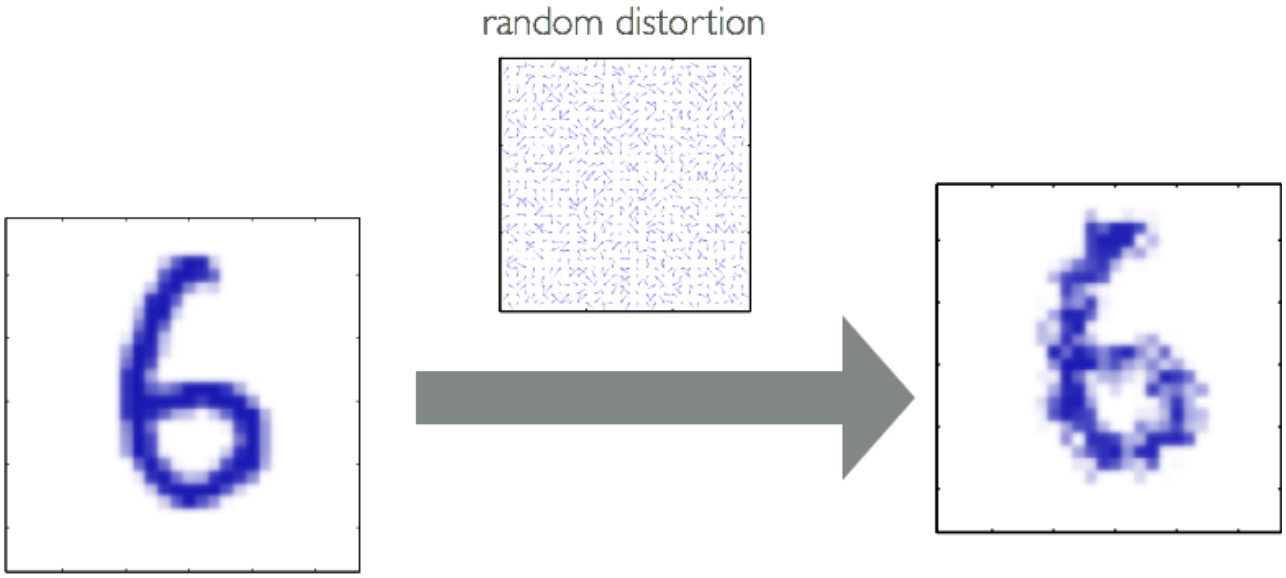
- However, it's easy to artificially generate data with such transformations
  - could use such data as additional training data
  - neural network can potentially learn to be invariant to such transformations

Generating Additional Examples



Elastic Distortions

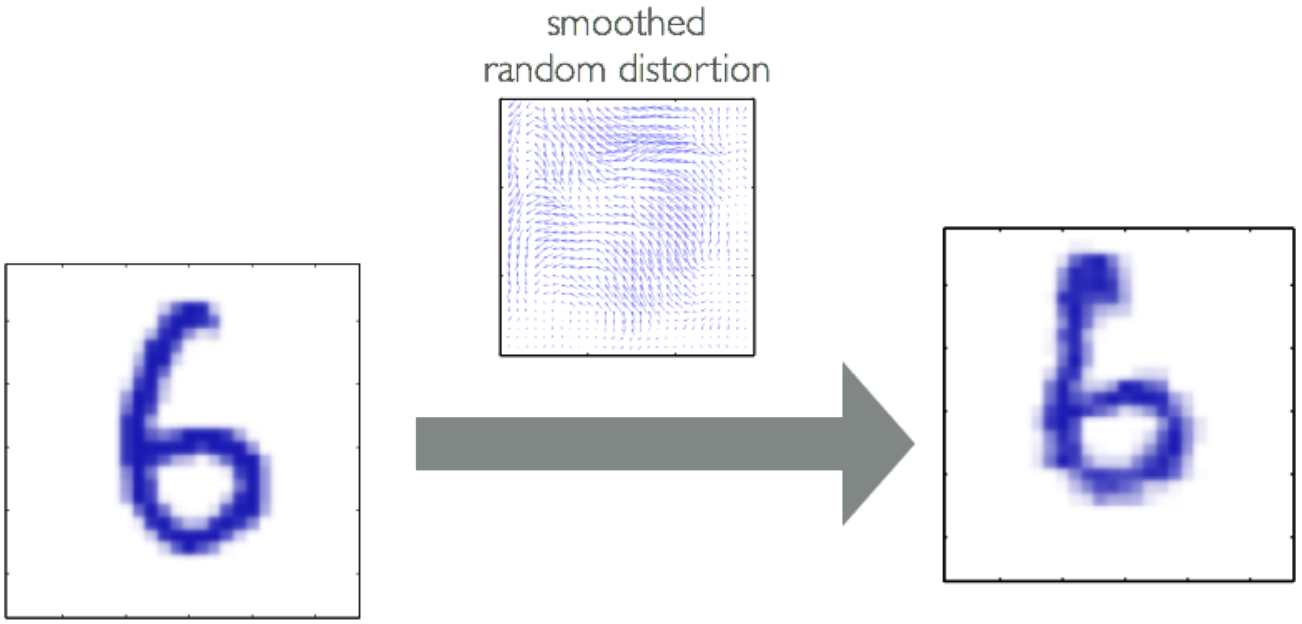
- Can add "elastic" deformations (useful in character recognition)
- We can do this by applying a "distortion field" to the image
  - a distortion field specifies where to displace each pixel value



Bishop's book

Elastic Distortions

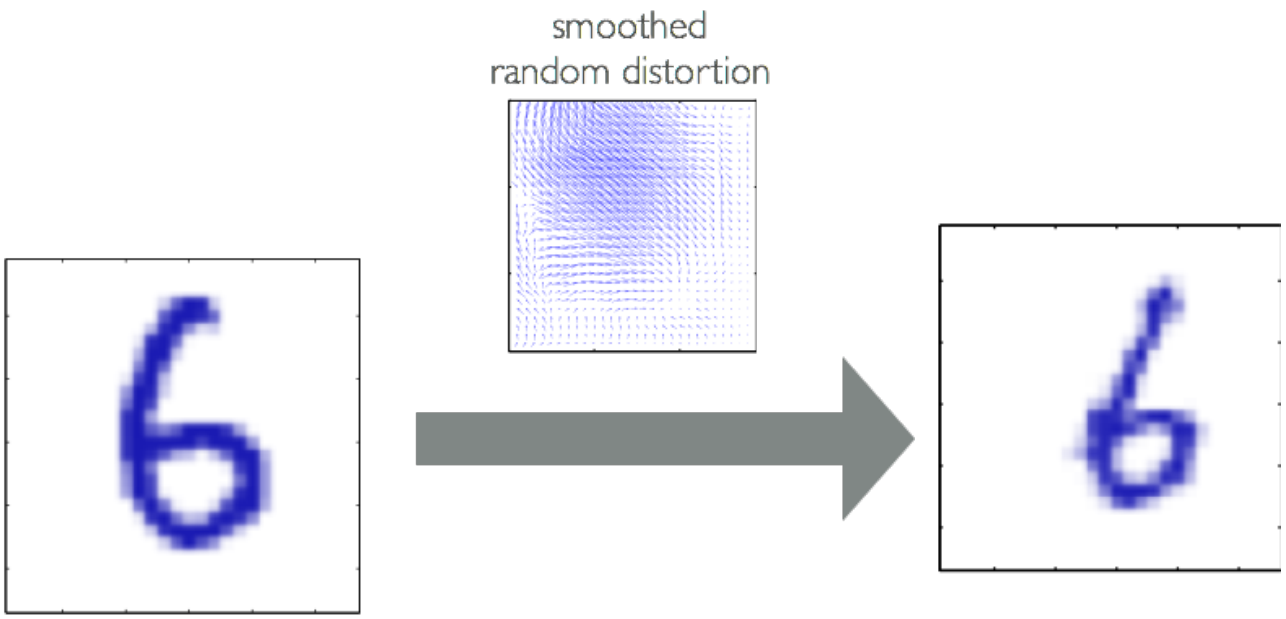
- Can add "elastic" deformations (useful in character recognition)
- We can do this by applying a "distortion field" to the image
  - a distortion field specifies where to displace each pixel value



Bishop's book

Elastic Distortions

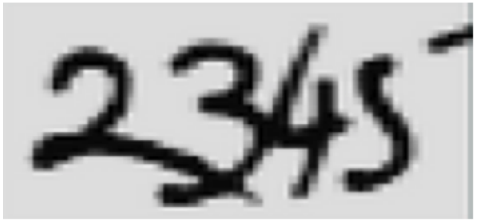
- Can add "elastic" deformations (useful in character recognition)
- We can do this by applying a "distortion field" to the image
  - a distortion field specifies where to displace each pixel value



Bishop's book

Conv Nets: Examples

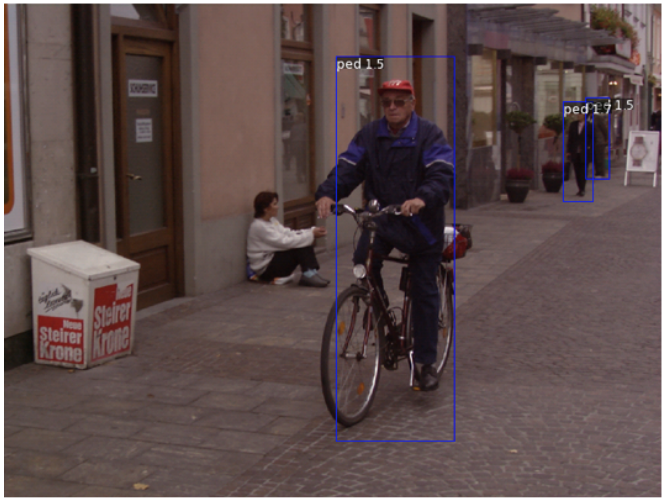
- Optical Character Recognition, House Number and Traffic Sign classification

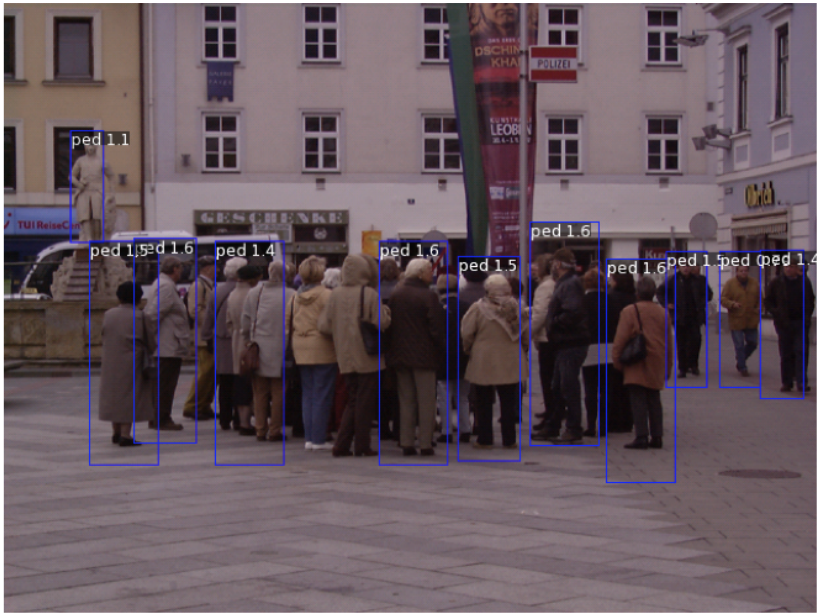


Ciresan et al. “MCDNN for image classification” CVPR 2012
Wan et al. “Regularization of neural networks using dropconnect” ICML 2013
Goodfellow et al. “Multi-digit nuber recognition from StreetView...” ICLR 2014
Jaderberg et al. “Synthetic data and ANN for natural scene text recognition” arXiv 2014

Conv Nets: Examples

- Pedestrian detection

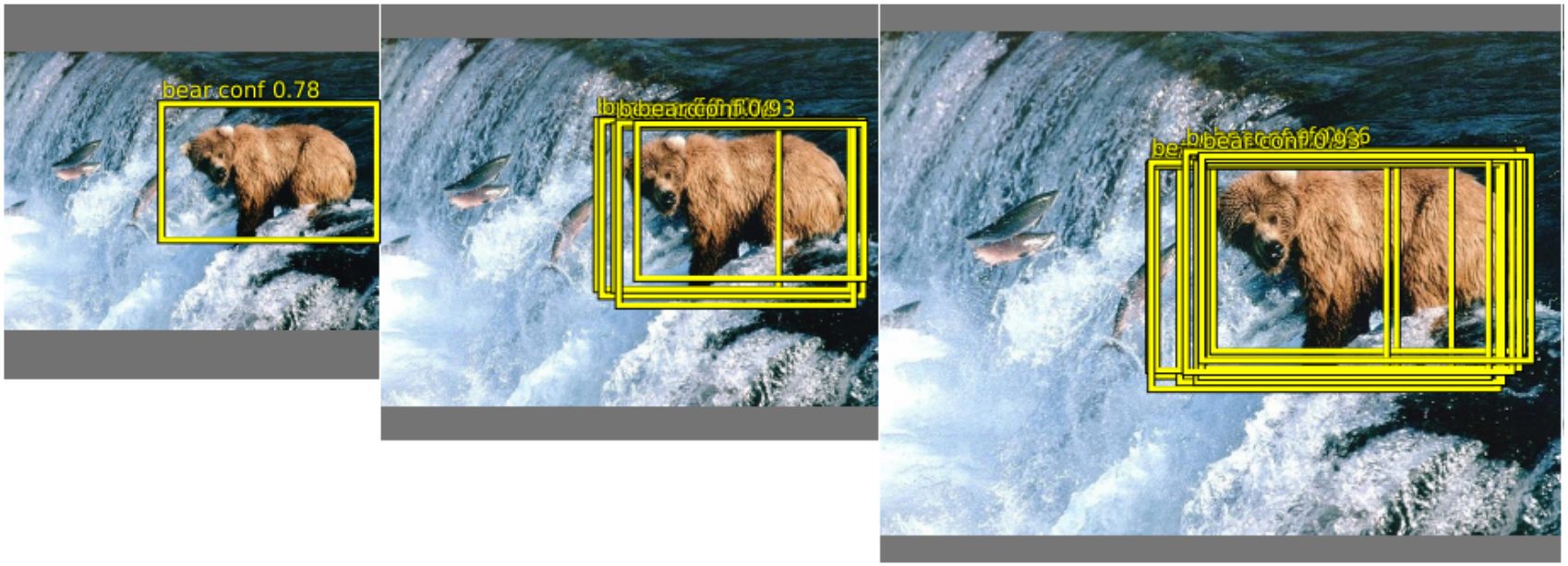




Sermanet et al. “Pedestrian detection with unsupervised multi-stage..” CVPR 2013

Conv Nets: Examples

- Object Detection

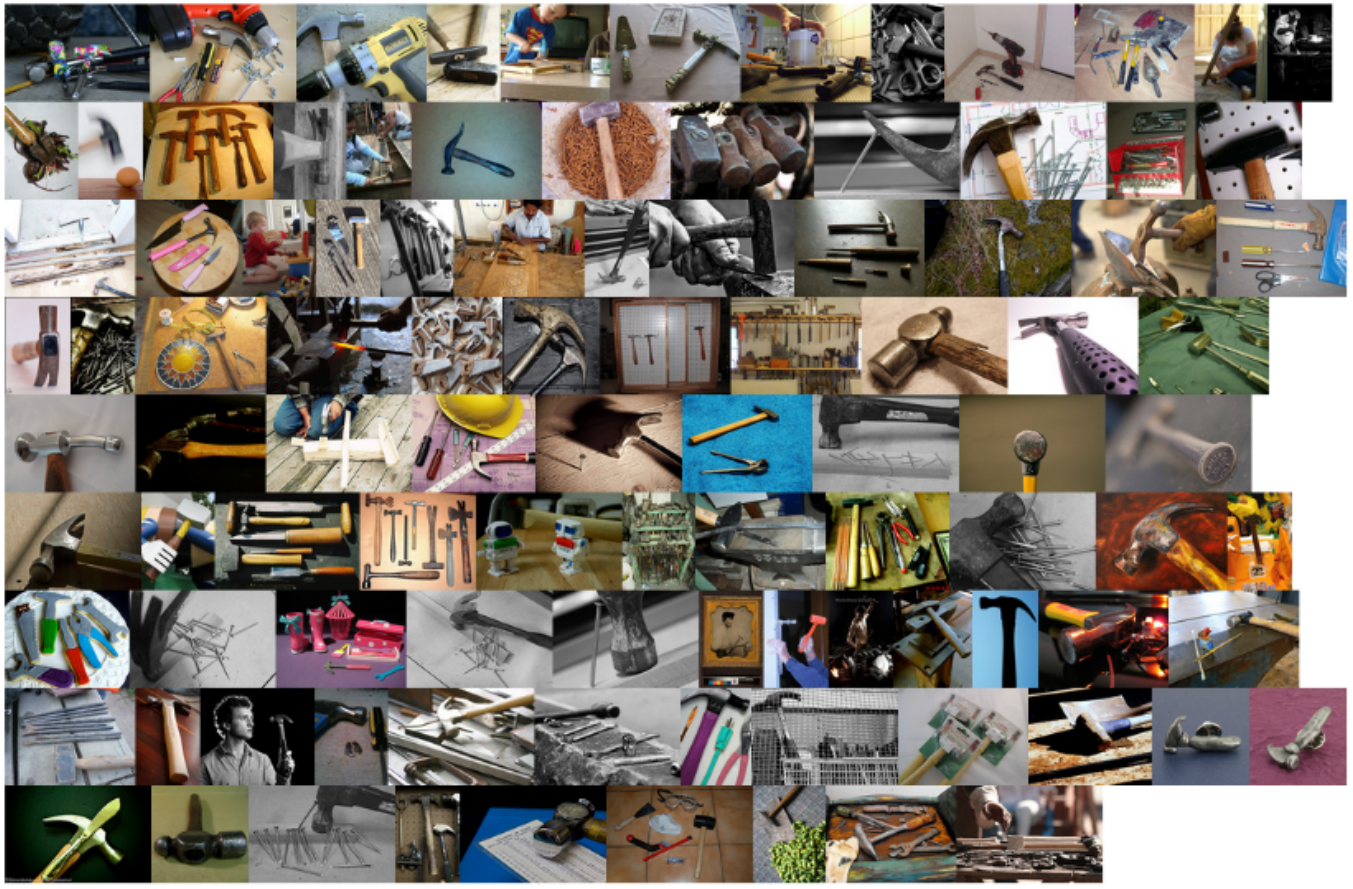


Sermanet et al. “OverFeat: Integrated recognition, localization” arxiv 2013
Girshick et al. “Rich feature hierarchies for accurate object detection” arxiv 2013
Szegedy et al. “DNN for object detection” NIPS 2013

ImageNet Dataset

- 1.2 million images, 1000 classes

Examples of Hammer

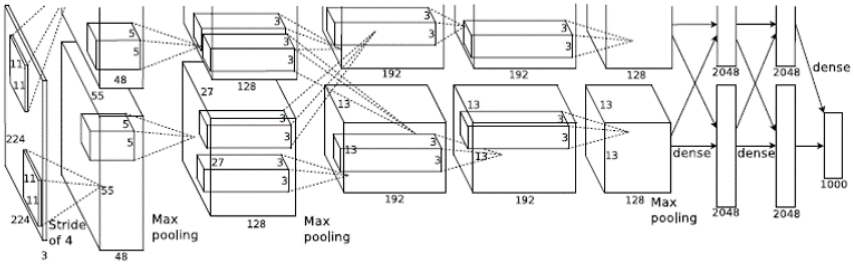


Deng et al. “Imagenet: a large scale hierarchical image database” CVPR 2009

Important Breakthroughs

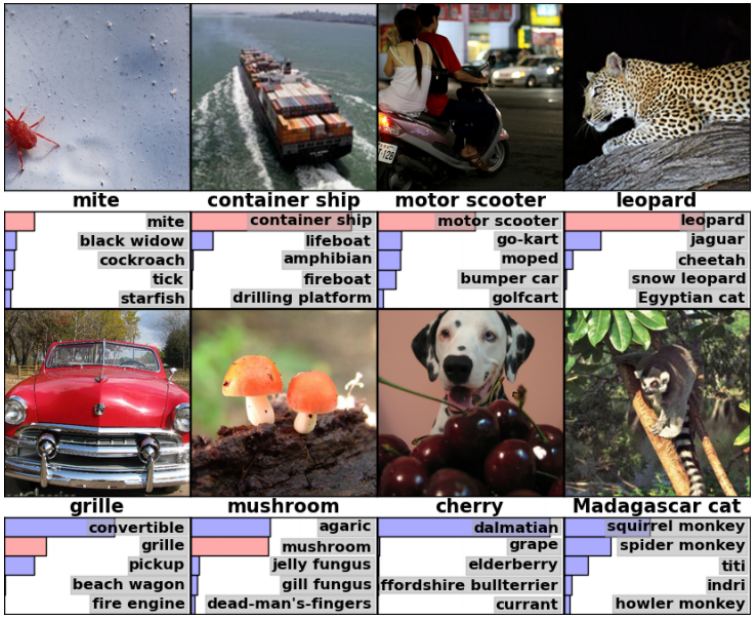
- Deep Convolutional Nets for Vision (Supervised)

  Krizhevsky, A., Sutskever, I. and Hinton, G. E., ImageNet Classification with Deep Convolutional Neural Networks, NIPS, 2012.



IMAGENET

1.2 million training images
1000 classes



Architecture

- How can we select the right architecture:
  - Manual tuning of features is now replaced with the manual tuning of architechtures

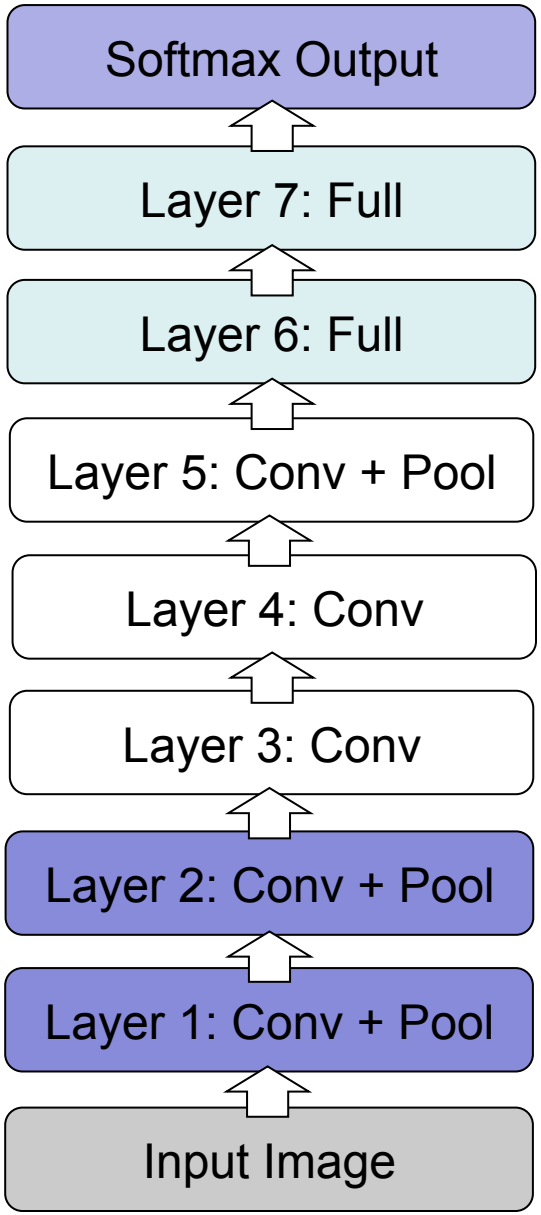
  - Depth
  - Width
  - Parameter count

How to Choose Architecture

- Many hyper-parameters:
  - Number of layers, number of feature maps
- Cross Validation
- Grid Search (need lots of GPUs)
- Smarter Strategies
  - Random search
  - Bayesian Optimization

AlexNet

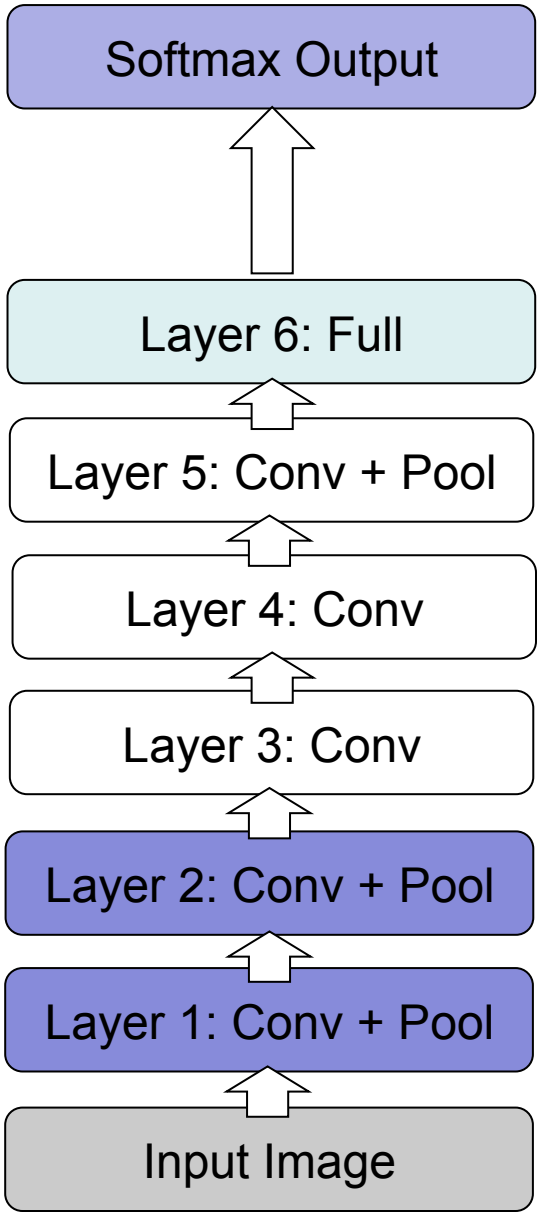
- 8 layers total
- Trained on Imagenet dataset [Deng et al. CVPR'09]
- 18.2% top-5 error



[From Rob Fergus' CIFAR 2016 tutorial]

AlexNet

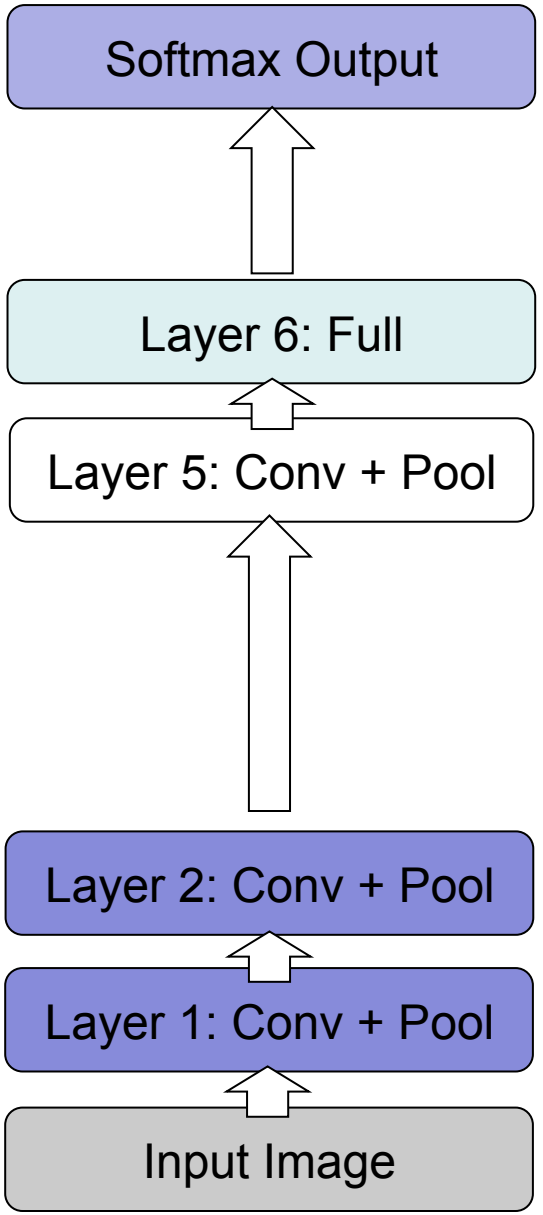
- Remove top fully connected layer 7
- Drop ~16 million parameters
- Only 1.1% drop in performance!



[From Rob Fergus' CIFAR 2016 tutorial]

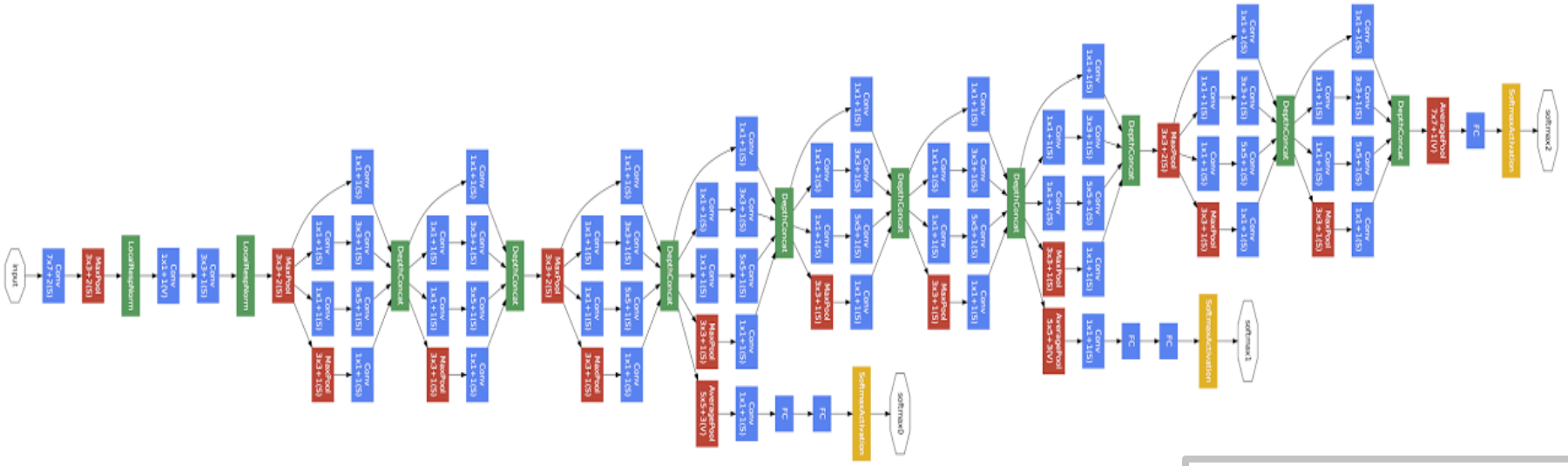
AlexNet

- Let us remove upper feature extractor layers and fully connected:
  - Layers 3,4, 6 and 7
- Drop ~50 million parameters
- **33.5 drop in performance!**
- Depth of the network is the key.



[From Rob Fergus' CIFAR 2016 tutorial]

GoogLeNet



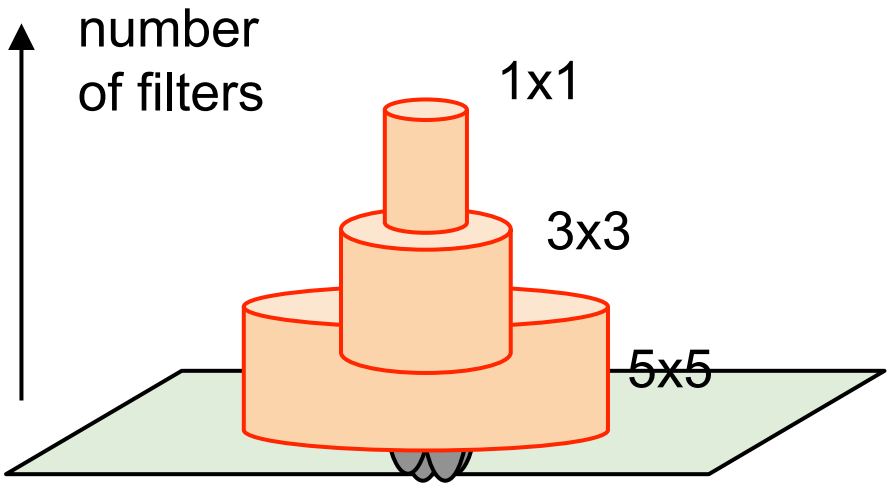
- 24 layer model that uses so-called inception module.

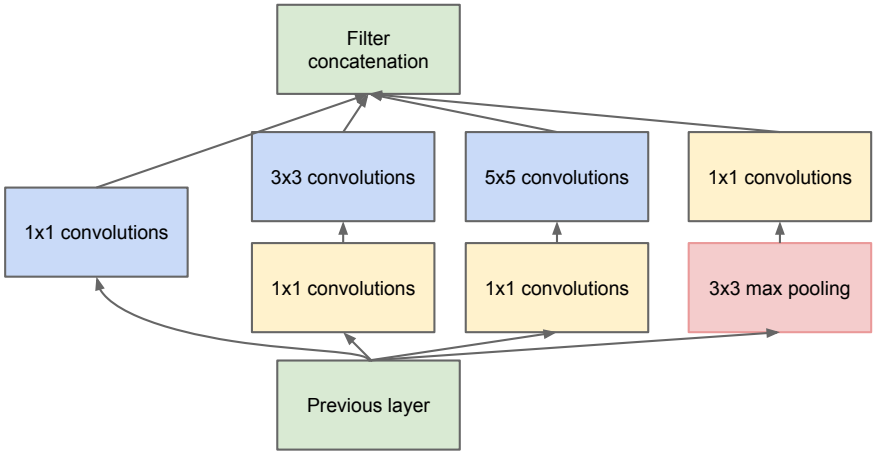
Convolution
Pooling
Softmax
Other

[Going Deep with Convolutions, Szegedy et al., arXiv:1409.4842, 2014]

GoogLeNet

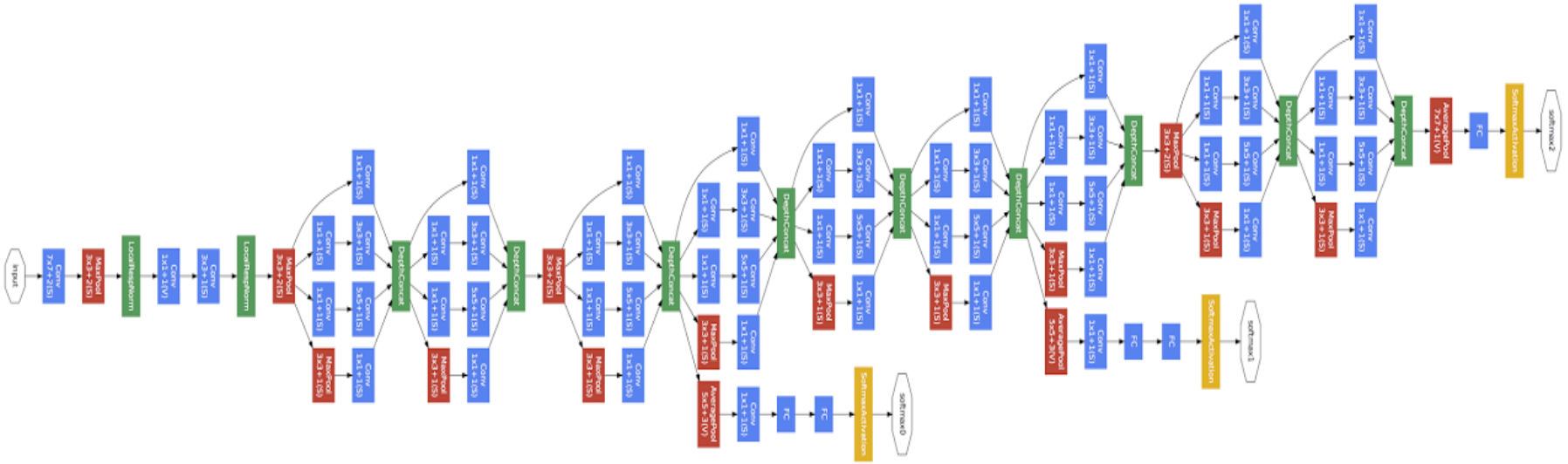
- GoogLeNet inception module:
  - Multiple filter scales at each layer
  - Dimensionality reduction to keep computational requirements down





[Going Deep with Convolutions, Szegedy et al., arXiv:1409.4842, 2014]

GoogLeNet

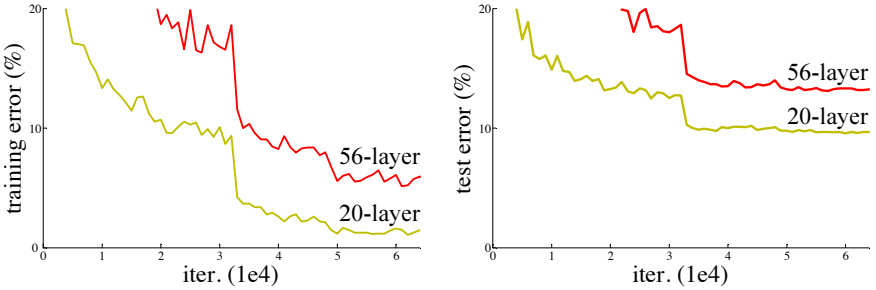


- Width of inception modules ranges from 256 filters (in early modules) to 1024 in top inception modules.
- Can remove fully connected layers on top completely
- Number of parameters is reduced to 5 million
- 6.7% top-5 validation error on Imagnet

[Going Deep with Convolutions, Szegedy et al., arXiv:1409.4842, 2014]

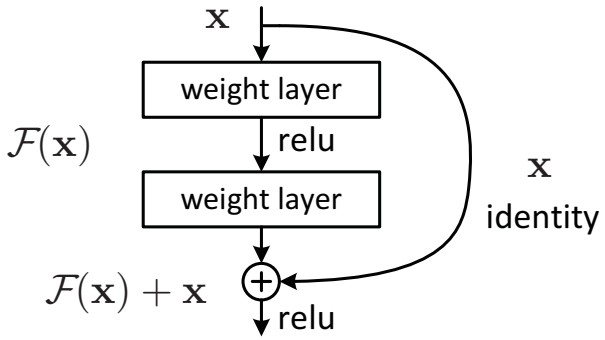
Residual Networks

Really, really deep convnets do not train well,
E.g. CIFAR10:



Key idea: introduce “pass through” into each layer

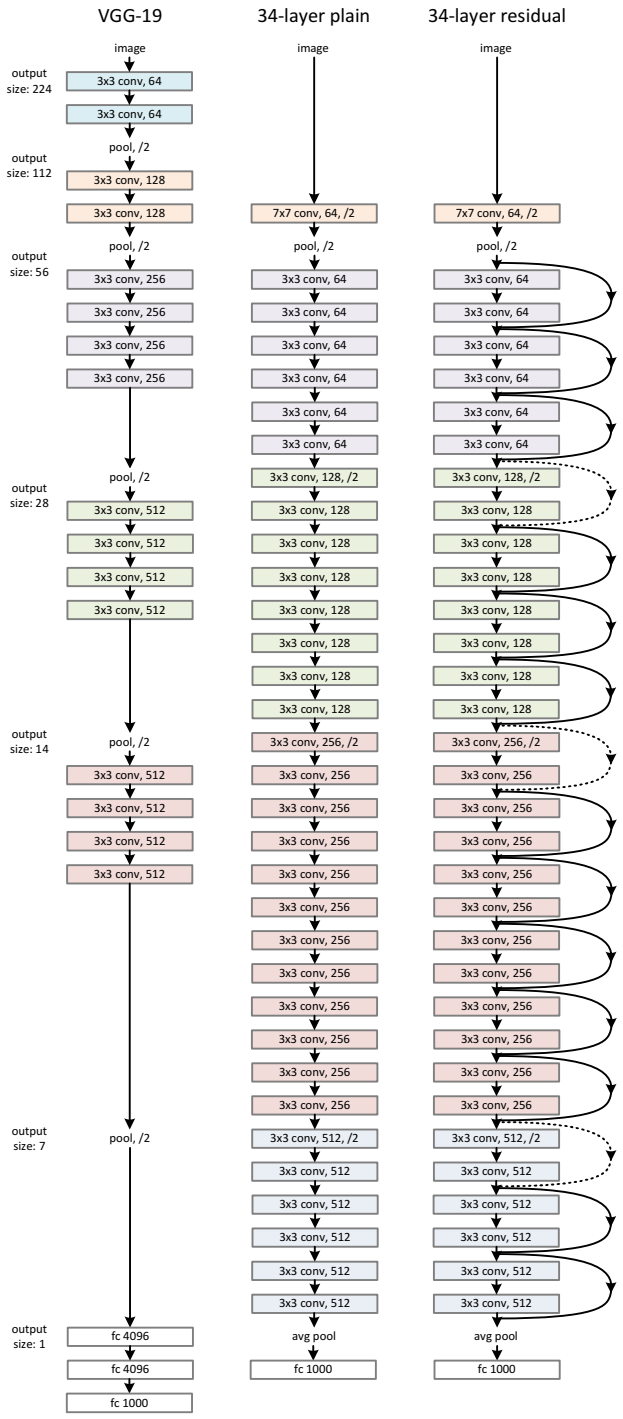
Thus only residual now needs to be learned



method	top-1 err.	top-5 err.
VGG [41] (ILSVRC’14)	-	8.43†
GoogLeNet [44] (ILSVRC’14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except † reported on the test set).

With ensembling, 3.57% top-5 test error on ImageNet



[He, Zhang, Ren, Sun, CVPR 2016]

Choosing the Architecture

- Task dependent
- Cross-validation
- [Convolution → pooling]* + fully connected layer
- The more data: the more layers and the more kernels
  - Look at the number of parameters at each layer
  - Look at the number of flops at each layer
- Computational resources

[From Marc'Aurelio Ranzato, CVPR 2014 tutorial]

Optimization Tricks

- SGD with momentum, batch-normalization, and dropout usually works very well

- Pick learning rate by running on a subset of the data
  - Start with large learning rate & divide by 2 until loss does not diverge
  - Decay learning rate by a factor of ~100 or more by the end of training

- Use ReLU nonlinearity

- Initialize parameters so that each feature across layers has similar variance. Avoid units in saturation.

[From Marc'Aurelio Ranzato, CVPR 2014 tutorial]

Improving Generalization

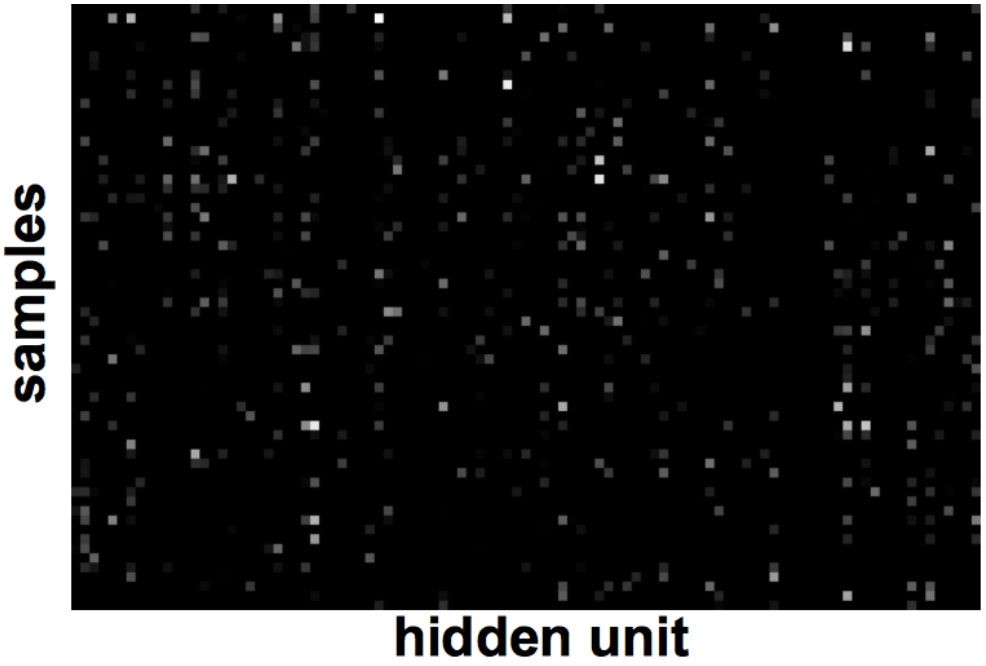
- Weight sharing (greatly reduce the number of parameters)
- Data augmentation (e.g., jittering, noise injection, etc.)
- Dropout
- Weight decay (L2, L1)
- Sparsity in the hidden units
- Multi-task (unsupervised learning)

[From Marc'Aurelio Ranzato, CVPR 2014 tutorial]

Visualization

- Check gradients numerically by finite differences

- Visualize features (feature maps need to be uncorrelated) and have high variance



- Good training: hidden units are sparse across samples

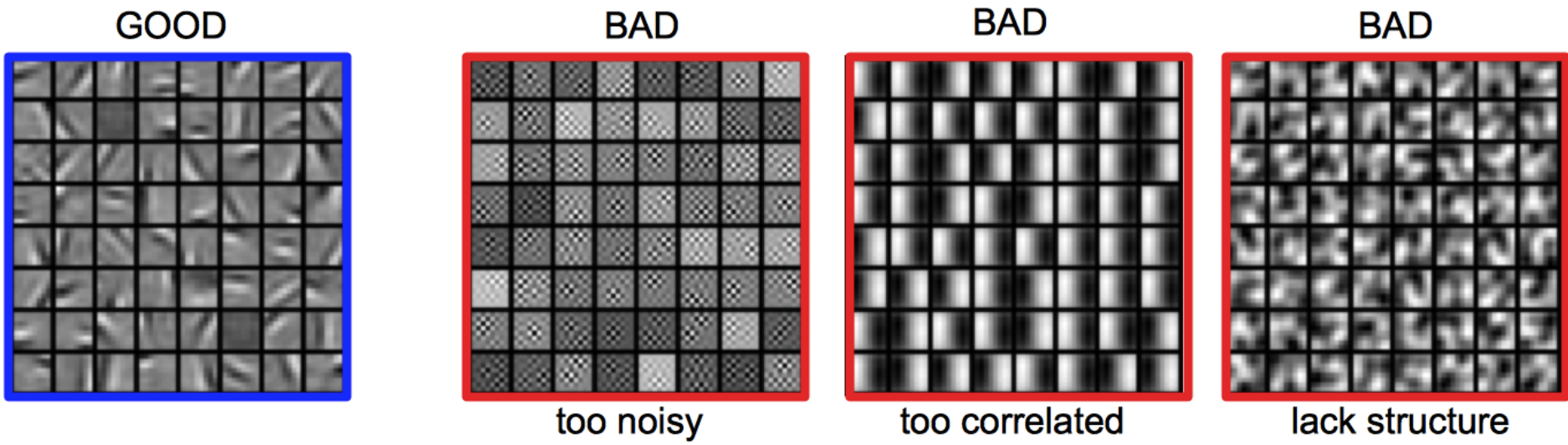
[From Marc'Aurelio Ranzato, CVPR 2014 tutorial]

Visualization

- Check gradients numerically by finite differences

- Visualize features (feature maps need to be uncorrelated) and have high variance

- Visualize parameters: learned features should exhibit structure and should be uncorrelated and are uncorrelated

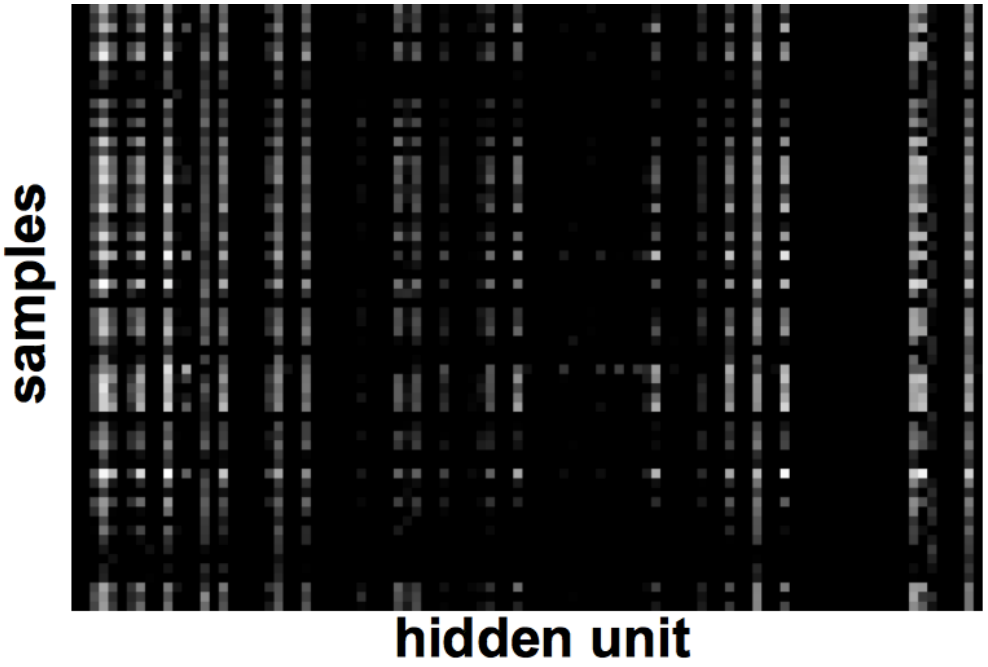


[From Marc'Aurelio Ranzato, CVPR 2014 tutorial]

Visualization

- Check gradients numerically by finite differences

- Visualize features (feature maps need to be uncorrelated) and have high variance



- Bad training: many hidden units ignore the input and/or exhibit strong correlations

[From Marc'Aurelio Ranzato, CVPR 2014 tutorial]

Visualization

- Check gradients numerically by finite differences

- Visualize features (feature maps need to be uncorrelated) and have high variance

- Visualize parameters: learned features should exhibit structure and should be uncorrelated and are uncorrelated

- Measure error on both training and validation set

- Test on a small subset of the data and check the error $\rightarrow$ 0.

[From Marc'Aurelio Ranzato, CVPR 2014 tutorial]

When it does not work

- Training diverges:
  - Learning rate may be too large → decrease learning rate
  - BPROP is buggy → numerical gradient checking
- Parameters collapse / loss is minimized but accuracy is low
  - Check loss function: Is it appropriate for the task you want to solve?
  - Does it have degenerate solutions?
- Network is underperforming
  - Compute flops and nr. params. → if too small, make net larger
  - Visualize hidden units/params → fix optimization
- Network is too slow
  - GPU,distrib. framework, make net smaller

[From Marc'Aurelio Ranzato, CVPR 2014 tutorial]

Supervised Learning

- Training time
  - Data:
    $\{\mathbf{x}^{(t)}, y^{(t)}\}$
  - Setting:
    $\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$

- Test time
  - Data:
    $\{\mathbf{x}^{(t)}, y^{(t)}\}$
  - Setting:
    $\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$

- Example: Classification, Regression

Unsupervised Learning

- Training time
  - Data:
    $\{\mathbf{x}^{(t)}\}$
  - Setting:
    $\mathbf{x}^{(t)} \sim p(\mathbf{x})$

- Test time
  - Data:
    $\{\mathbf{x}^{(t)}\}$
  - Setting:
    $\mathbf{x}^{(t)} \sim p(\mathbf{x})$

- Example: Distribution Estimation, Dimensionality Reduction

Semi-Supervised Learning

- Training time
  - Data:
    $\{\mathbf{x}^{(t)}, y^{(t)}\}$
    $\{\mathbf{x}^{(t)}\}$
  - Setting:
    $\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$
    $\mathbf{x}^{(t)} \sim p(\mathbf{x})$

- Test time
  - Data:
    $\{\mathbf{x}^{(t)}, y^{(t)}\}$
    $\{\mathbf{x}^{(t)}\}$
  - Setting:
    $\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$
    $\mathbf{x}^{(t)} \sim p(\mathbf{x})$

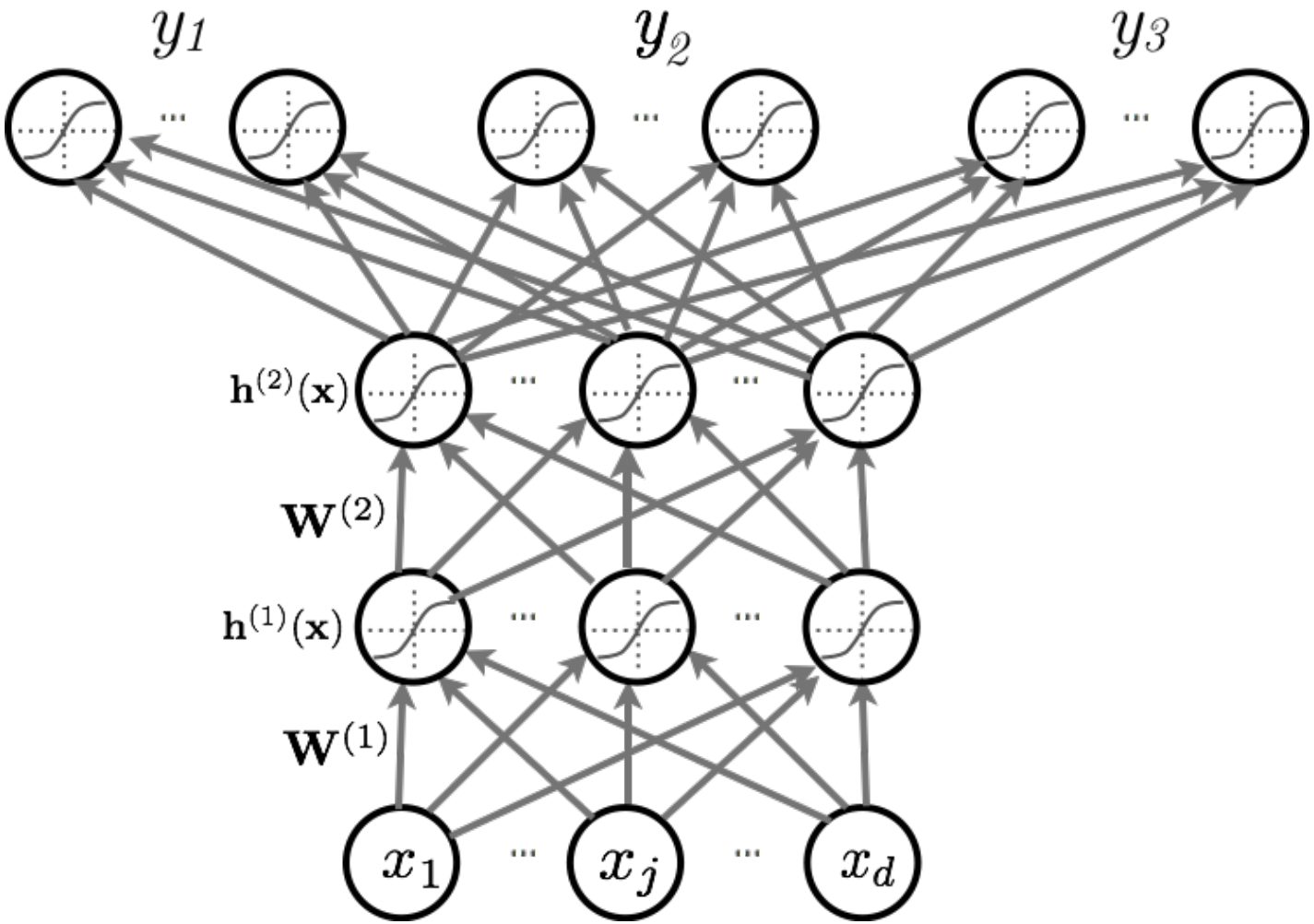
Multi-Task Learning

- Training time
  - Data:
    $\{\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)}\}$
  - Setting:
    $\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)} \sim p(\mathbf{x}, y_1, \ldots, y_M)$

- Test time
  - Data:
    $\{\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)}\}$
  - Setting:
    $\mathbf{x}^{(t)}, y_1^{(t)}, \ldots, y_M^{(t)} \sim p(\mathbf{x}, y_1, \ldots, y_M)$

- Example: object recognition in images with multiple objects

Multi-Task Learning



Structured Output Prediction

- Training time
  - Data:

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

Data of arbitrary structure (vector, sequence, graph).

  - Setting:

$$\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$$

- Test time
  - Data:

$$\{\mathbf{x}^{(t)}, y^{(t)}\}$$

Data of arbitrary structure (vector, sequence, graph).

  - Setting:

$$\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$$

- Example: Image caption generation, machine translation

One-Shot Learning

- Training time
  - Data:
    $\{\mathbf{x}^{(t)}, y^{(t)}\}$
  - Setting:
    $\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$
    $y^{(t)} \in \{1, \ldots, C\}$

- Example: recognizing a person based on a single picture of him/her

- Test time
  - Data:
    $\{\mathbf{x}^{(t)}, y^{(t)}\}$
  - Setting:
    $\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$
    $y^{(t)} \in \{C+1, \ldots, C+M\}$

Additional data: A single labeled example from each of the M new classes

Zero-Shot Learning

- Training time
  - Data:
    $\{\mathbf{x}^{(t)}, y^{(t)}\}$
  - Setting:
    $\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$
    $y^{(t)} \in \{1, \ldots, C\}$

Additional data: Description vector $\mathbf{z}_c$ of each of the C classes

- Test time
  - Data:
    $\{\mathbf{x}^{(t)}, y^{(t)}\}$
  - Setting:
    $\mathbf{x}^{(t)}, y^{(t)} \sim p(\mathbf{x}, y)$
    $y^{(t)} \in \{C+1, \ldots, C+M\}$

Additional data: description vector $\mathbf{z}_c$ of each of the new M classes

Can you solve Zero-Shot Problem?

Description [edit]

These birds have yellow underparts, blue-grey upperparts and pink legs; they also have yellow eye-rings and thin, pointed bills. Adult males have black foreheads and black necklaces. Females and immatures have faint grey necklaces. They have yellow “spectacles” round the eyes.

The Canada warbler is the host to the parasite *Apororhynchus amphistomi*.[2]



Can you solve Zero-Shot Problem?

Description [edit]

These birds have yellow underparts, blue-grey upperparts and pink legs; they also have yellow eye-rings and thin, pointed bills. Adult males have black foreheads and black necklaces. Females and immatures have faint grey necklaces. They have yellow "spectacles" round the eyes.

The Canada warbler is the host to the parasite *Apororhynchus amphistomi*.[2]

Canada Warbler

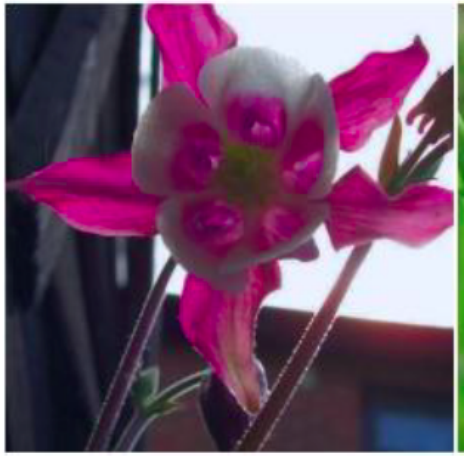
Yellow Warbler

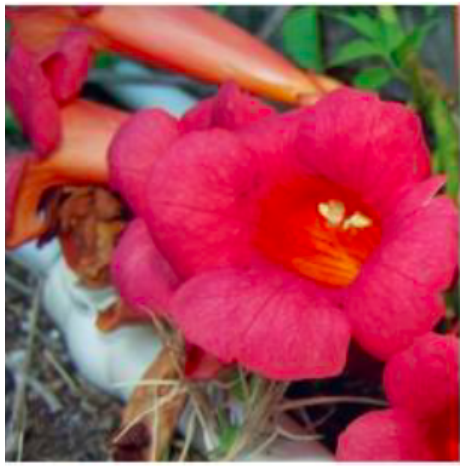
Sharpe-tailed Sparrow

Can you solve Zero-Shot Problem?

Description [edit]

Fritillaries often have nodding, bell- or cup-shaped flowers, and the majority are spring-flowering. Certain species have flowers that emit disagreeable odors. The scent of *Fritillaria imperialis* has been called "rather nasty", while that of *F. agrestis*, known commonly as stink bells, is reminiscent of dog droppings.[6] On the other hand, *F. striata* has a sweet fragrance.[6]





The Model

- Consider binary one vs. all classifier for class c:

$$\hat{y}_c = w_c^\top x$$

Weight vector for a particular class c

- How can we deal with previously unseen classes using this standard formulation?

Ba, Swersky, Fidler, Salakhutdinov, ICCV 2015

The Model

- Consider binary one vs. all classifier for class c:

$$\hat{y}_c = w_c^\top x$$

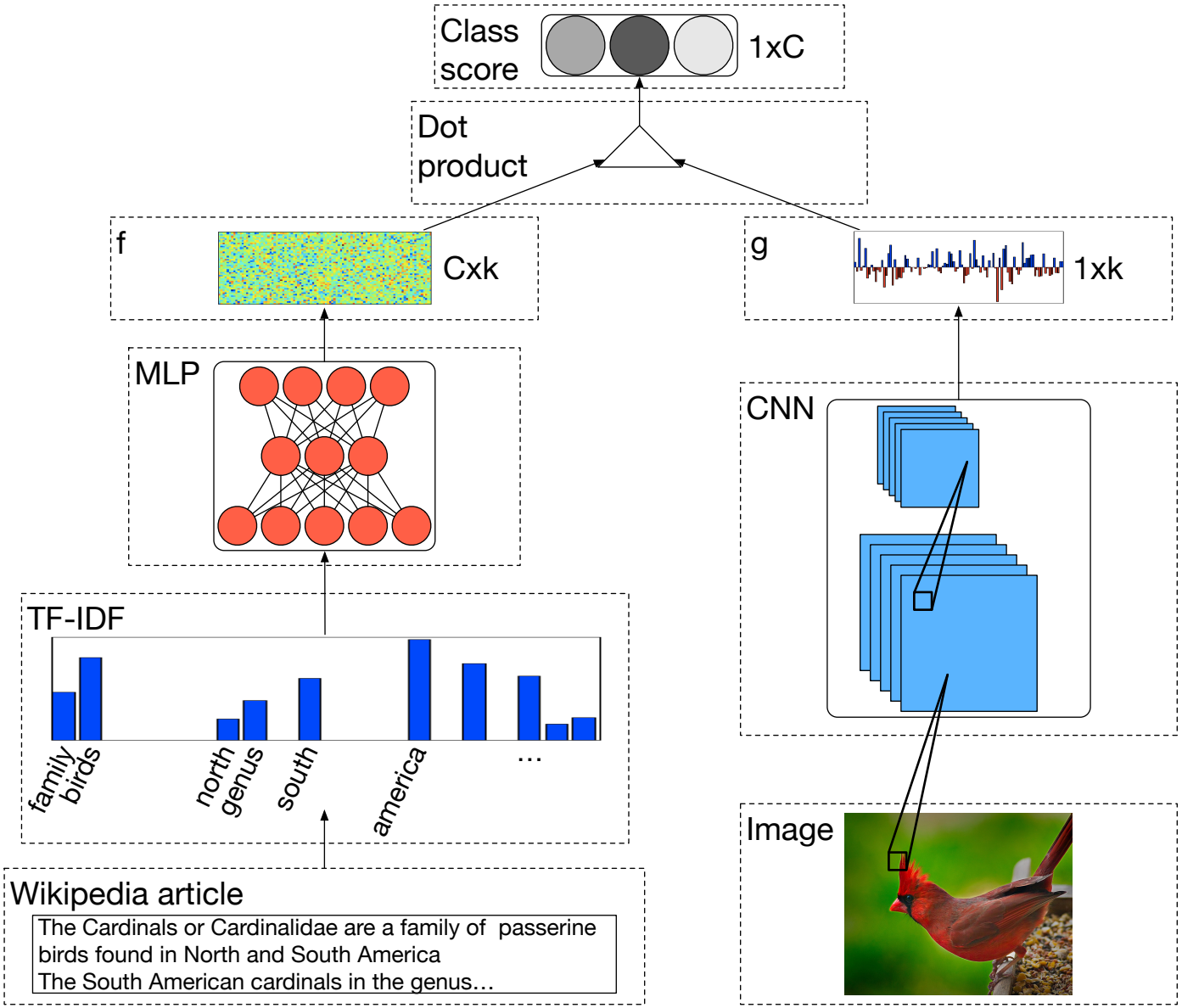
- Assume that we are given an addition text feature $t_c \in \mathbb{R}^p$.
- **Simple idea**: Instead of learning a static weight vector w_c , the text feature can be used to predict the weights:

$$w_c = f_t(t_c),$$

where $f_t : \mathbb{R}^p \mapsto \mathbb{R}^d$ is a mapping that transforms the text features to the visual image feature space.

- We can use this idea to predict the output weights of a classifier (both the fully connected and convolutional layers of a CNN).

Model Architecture

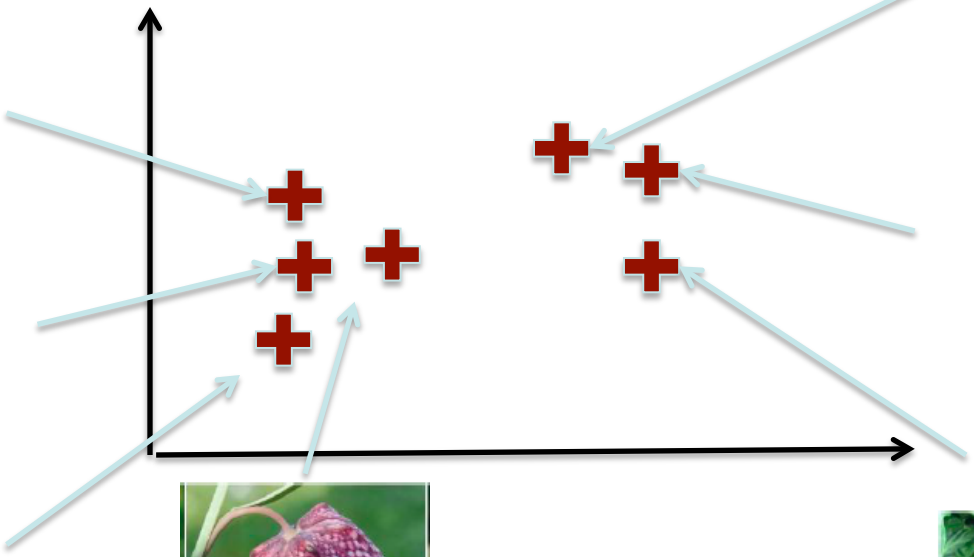


Alternative View

Fritillary

Fritillaria is a genus of about 100 species of bulbous plants in the family Liliaceae, native to temperate regions of the Northern Hemisphere. The name is derived from the Latin term for a dice-box (fritillus), and probably refers to the checkered pattern, frequently of chocolate-brown and greenish yellow, that is common to many species' flowers. Collectively, the genus is known in English as fritillaries; some North American species are called mission bells.
They often have nodding, bell- or cup-shaped flowers, and the majority are spring-flowering. Most species' flowers have a rather disagreeable scent, often referred to as "foxy," like feces or wet fur. The Scarlet Lily Beetle (Lilioceris lilii) eats fritillaries, and may become a pest where these plants are grown in gardens.
Several species (such as F. cirrhosa and F. verticillata) are used in traditional Chinese cough remedies. They are listed as chuan bÊi (Chinese: ??) or zhÊ bÊi (Chinese: ??), respectively, and are often in formulations combined with extracts of Loquat (Eriobotrya japonica). F. verticillata bulbs are also traded as bÊi mu or, in Kampo, beimo (Chinese/Kanji: ??, Katakana: ???). F. thunbergii is contained in the standardized Chinese herbal preparation HealthGuard T18, taken against hyperthyroidism.

Joint Semantic
Feature space



Corn Poppy

Papaver rhoeas (common names include corn poppy, corn rose, field poppy, Flanders poppy, red poppy, red weed, coquelicot, and, due to its odour, which is said to cause them, as headache and headwark) is a species of flowering plant in the poppy family, Papaveraceae. This poppy, a native of Europe, is notable as an agricultural weed (hence the "corn" and "field") and as a symbol of fallen soldiers.
P. rhoeas sometimes is so abundant in agricultural fields that it may be mistaken for a crop. The only species of Papaveraceae grown as a field crop on a large scale is Papaver somniferum, the opium poppy.

The plant is a variable annual, forming a long-lived soil seed bank that can germinate when the soil is disturbed. In the northern hemisphere it generally flowers in late spring, but if the weather is warm enough other flowers frequently appear at the beginning of autumn. The flower is large and showy, with four petals that are vivid red, most commonly with a black spot at their base. Like many other species of Papaver, it exudes a white latex when the tissues are broken.

- Minimize the cross-entropy or hinge-loss objective.

Problem Setup

• The training set contains N images $x \in R^d$ and their associated class labels $l \in \{1, ..., C\}$. There are C distinct class labels.

• During test time, we are given additional n_0 number of the previously unseen classes, such that $l_{test} \in \{1, ..., C, ...C+n_0\}$.

• Our goal is to predict previously unseen classes and perform well on the previously seen classes.

- Interpretation: Learning a good similarity kernel between images and encyclopedia articles .

Caltech UCSD Bird and Oxford Flower Datasets

• The CUB200-2010 contains 6,033 images from 200 bird species (about 30 images per class).

• There is one Wikipedia article associated with each bird class (200 articles in total). The average number of words per articles is 400.

• Out of 200 classes, 40 classes are defined as unseen and the remaining 160 classes are defined as seen.

• The Oxford Flower-102 dataset contains 102 classes with a total of 8189 images.

• Each class contains 40 to 260 images. 82 flower classes are used for training and 20 classes are used as unseen during testing.

Results: Area Under ROC

The CUB200-2010 Dataset

Learning Algorithm	Unseen	Seen	Mean
DA [Elhoseiny, et.al., 2013]	0.59	-	-
DA (VGG features)	0.62	-	-
Ours (fc)	**0.82**	0.96	0.934
Ours (conv)	0.73	0.96	0.91
Ours (fc+conv)	0.80	**0.987**	**0.95**

M. Elhoseiny, B. Saleh, and A. Elgammal, ICCV, 2013.

Results: Area Under ROC

The Oxford Flower Dataset

Learning Algorithm	Unseen	Seen	Mean
DA [Elhoseiny, et.al.]	0.62	-	-
GPR+DA [Elhoseiny, et.al.]	0.68	-	-
Ours (fc)	0.70	0.987	0.90
Ours (conv)	0.65	0.97	0.85
Ours (fc+conv)	**0.71**	**0.989**	**0.93**

M. Elhoseiny, B. Saleh, and A. Elgammal, ICCV, 2013.

Attribute Discovery

Scarlet Tanager

The Scarlet Tanager (Piranga olivacea) is a medium-sized American songbird. Formerly placed in the tanager family (Thraupidae), it and other members of its genus are now classified in the cardinal family (Cardinalidae). The species's plumage and vocalizations are similar to other members of the cardinal family.

Adults have pale stout smooth bills. Adult males are bright red with black wings and tail; females are yellowish on the underparts and olive on top, with olive-brown wings and tail. The adult male's winter plumage is similar to the female's, but the wings and tail remain darker. Young males briefly show a more complex variegated plumage intermediate between adult males and females. It apparently was such a specimen that was first scientifically described.{Citation needed|date=July 2010} Hence the older though somewhat confusing specific epithet olivacea ("the olive-colored one") is used rather than erythromelas ("the red-and-black one"), as had been common throughout the 19th century.

Word sensitivities of unseen classes

- Tanagers
- Scarlet
- Tanagers
- Thraupidae
- Cardinalidae

Nearest Neighbors

Scarlet Tanager

Summer Tanager

Vermillion Flycatcher

Brown Thrasher

• Wikipedia article for each class is projected onto its feature space and the nearest image neighbors from the test-set are shown.

Attribute Discovery

Bearded Iris

Irises are perennial plants, growing from creeping rhizomes (rhizomatous irises) or, in drier climates, from bulbs (bulbous irises). They have long, erect flowering stems which may be simple or branched, solid or hollow, and flattened or have a circular cross-section. The rhizomatous species usually have 3–10 basal sword-shaped leaves growing in dense clumps. The bulbous species have cylindrical, basal leaves.

Word sensitivities of unseen classes

- rhizome
- freezing
- iris
- depth
- compost

Nearest Neighbors

Bearded Iris

Corn Poppy

Grape Hyacinth

Giant White Arum Lily

• Wikipedia article for each class is projected onto its feature space and the nearest image neighbors from the test-set are shown.