Theories of Deep Learning

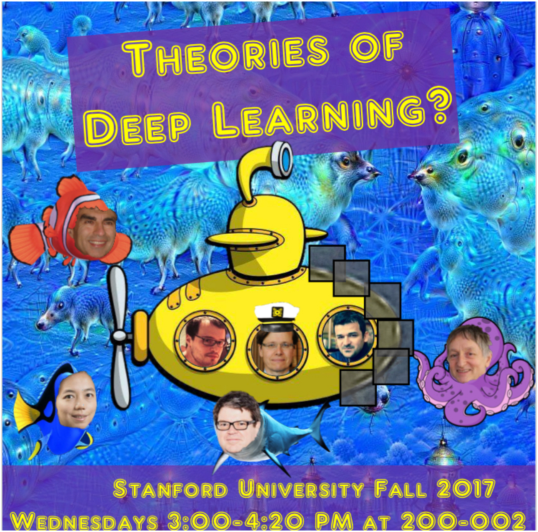
Lecture 02

Donoho, Monajemi, **Papyan**

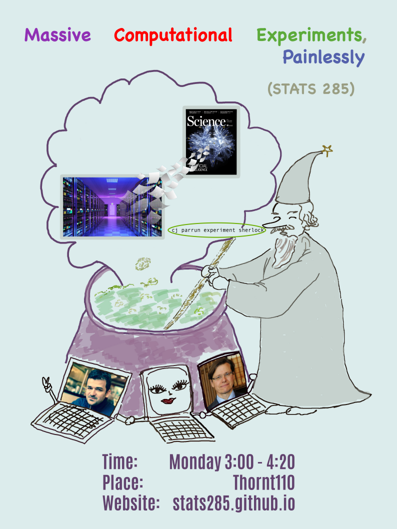
Department of Statistics
Stanford

Oct. 4, 2017

Stats 385 Fall 2017



Stats 285 Fall 2017



Course info

- Wed 3:00-4:20 PM in 200-002
- Sept 27 - Dec 6 (10 Weeks)
- Website: `http://stats385.github.io`
- @stats385
- Instructors:
  - David Donoho

Email	`donoho@stanxxx.edu`
Office hours	Mon/Wed 1PM in Sequoia 128

  - Hatef Monajemi

Email	`monajemi@stanxxx.edu`
Office hours	Mondays, 11:00 AM in Sequoia 216
Twitter	@hatefmnj

  - Vardan Papyan

Email	`papyan@stanxxx.edu`
Office hours	TBD

Reminders

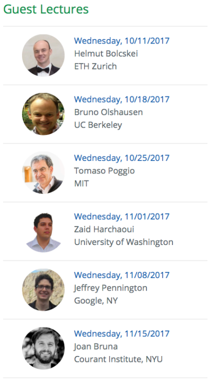
- Weekly guest lectures
- Associated abstracts, readings
- Projects
- Course Website: `http://stats385.github.io`
  - Each Week's Speaker
  - Readings (Links to Selected)
  - Announcements
  - Lecture Slides
- Stanford *Canvas* site
  - Readings (Incl. Copyrighted)
  - Announcements
  - Lecture Slides
  - Chat

Basic Information about Deep Learning

- Chris Manning:
  `http://web.stanford.edu/class/cs224n/`
- Pal Sujit's NLP tutorial:
  `https://github.com/sujitpal/eeap-examples`
- Andrew Ng's `deeplearning.ai`
- CS231n course website: `http://cs231n.github.io`
- PyTorch Tutorial (All kinds of examples):
  `http://pytorch.org/tutorials/`

- Books:
  - *Deep Learning*, Goodfellow, Bengio, Courville; 2016.
  - *Neural Networks and Deep Learning* Michael Nielsen
    `http://neuralnetworksanddeeplearning.com`
  - Many O'Reilly Books
  - `http://deeplearning.net/reading-list/`
  - Many NIPS Papers.

A Look Ahead: `https://stats385.github.io`



Next Two Lectures:

Wed Oct 11	Helmut Boelcskei	ETH Zuerich
Wed Oct 18	Ankit Patel	Rice

Wed Oct 11 Helmut Boelsckei

Readings for this lecture

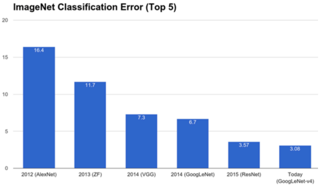
1. A mathematical theory of deep convolutional neural networks for feature extraction
2. Energy propagation in deep convolutional neural networks
3. Discrete deep feature extraction: A theory and new architectures
4. Topology reduction in deep convolutional feature extraction networks

Possibly also of interest

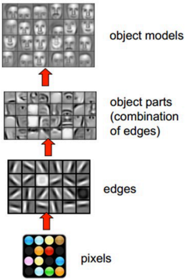
- S. Mallat, *Understanding Deep Convolutional Networks* Phil. Trans. Roy. Soc. 2017
- Mallat, Stéphane. "Group invariant scattering." Communications on Pure and Applied Mathematics 65, no. 10 (2012): 1331-1398

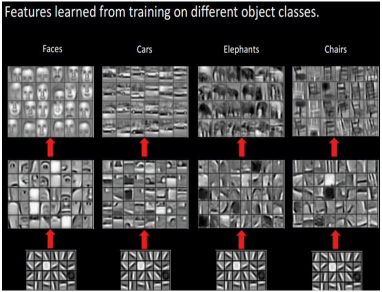
Lecture 1, in review

Global Economy → Computing → Deep Learning



Lecture 2, in overview





ImageNet dataset

- 14,197,122 labeled images
- 21,841 classes
- Labeling required more than a year of human effort via Amazon Mechanical Turk

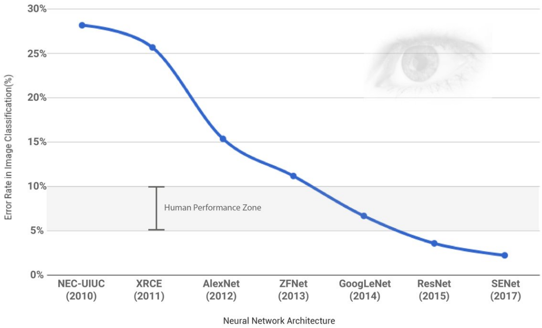
IMAGENET

The Common Task Framework

- Crucial methodology driving predictive modeling's success
- An instance has the following ingredients:
  - Training dataset
  - Competitors whose goal is to learn a predictor from the training set
  - Scoring referee

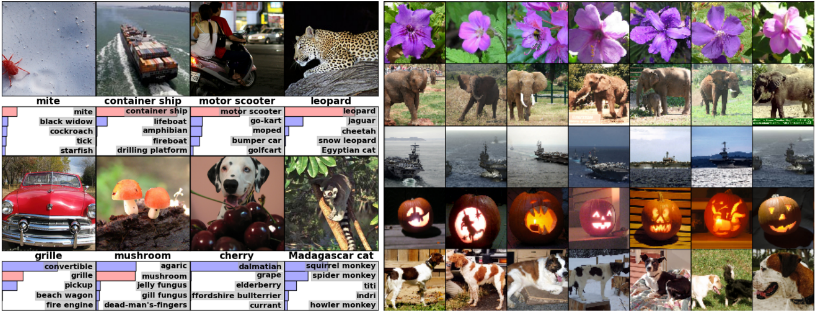
Instance of Common Task Framework, 1

- ImageNet (subset):
  - 1.2 million training images
  - 100,000 test images
  - 1000 classes
- ImageNet large-scale visual recognition Challenge



source: https://www.linkedin.com/pulse/must-read-path-breaking-papers-image-classification-muktabh-mayank

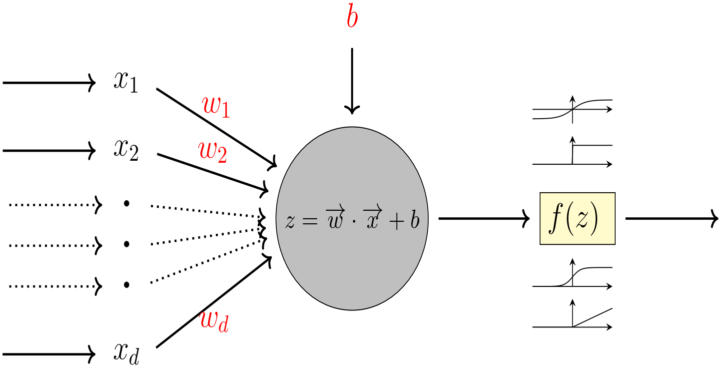
Instance of Common Task Framework, 2



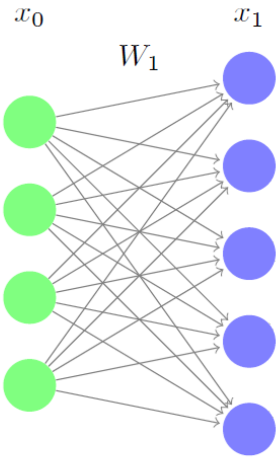
Source: [Krizhevsky et al., 2012]

Perceptron, the basic block

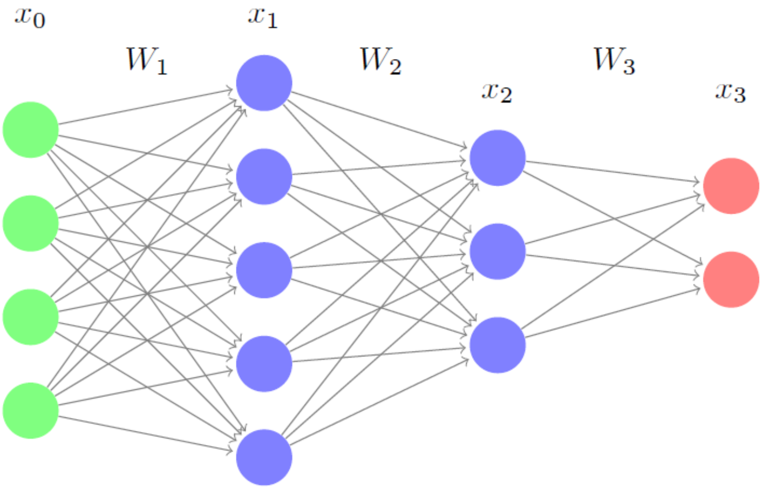
- Invented by Frank Rosenblatt (1957)



Single-layer perceptron



Multi-layer perceptron



Forward pass

- Cascade of repeated [linear operation followed by coordinatewise nonlinearity]'s
- Nonlinearities: sigmoid, hyperbolic tangent, (recently) ReLU.

Algorithm 1 Forward pass

Input: x_0
Output: x_L

1: **for** $\ell = 1$ to L **do**
2: $\quad x_\ell = f_\ell(W_\ell x_{\ell-1} + b_\ell)$
3: **end for**

Training neural networks

- Training examples $\{x_0^i\}_{i=1}^n$ and labels $\{y^i\}_{i=1}^n$
- Output of the network $\{x_L^i\}_{i=1}^m$
- Objective

$$J(\{W_l\},\{b_l\}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\|y^i - x_L^i\|_2^2 \tag{1}$$

- Gradient descent

$$W_l = W_l - \eta\frac{\partial J}{\partial W_l}$$
$$b_l = b_l - \eta\frac{\partial J}{\partial b_l}$$

: In practice: use Stochastic Gradient Descent (SGD)

back-propagation – derivation

derivation from LeCun et al. 1988

Given n training examples $(I_i, y_i) \equiv$ (input,target) and L layers

- Constrained optimization

$$\begin{array}{ll} \min\limits_{W,x} & \sum_{i=1}^{n} \|x_i(L) - y_i\|_2 \\ \text{subject to} & x_i(\ell) = f_\ell\Big[W_\ell x_i\left(\ell - 1\right)\Big], \\ & i = 1, \ldots, n, \quad \ell = 1, \ldots, L,\ x_i(0) = I_i \end{array}$$

- Lagrangian formulation (Unconstrained)

$$\min_{W,x,B} \mathcal{L}(W, x, B)$$

$$\mathcal{L}(W, x, B) = \sum_{i=1}^{n} \left\{ \|x_i(L) - y_i\|_2^2 + \sum_{\ell=1}^{L} B_i(\ell)^T \Big(x_i(\ell) - f_\ell\Big[W_\ell x_i\left(\ell - 1\right)\Big]\Big) \right\}$$

http://yann.lecun.com/exdb/publis/pdf/lecun-88.pdf

back-propagation – derivation

- $\frac{\partial \mathcal{L}}{\partial B}$

Forward pass

$$x_i(\ell) = f_\ell\Big[\underbrace{W_\ell x_i\,(\ell - 1)}_{A_i(\ell)}\Big] \quad \ell = 1, \ldots, L, \quad i = 1, \ldots, n$$

- $\frac{\partial \mathcal{L}}{\partial x}, z_\ell = [\nabla f_\ell]B(\ell)$

Backward (adjoint) pass

$$z(L) = 2\nabla f_L\Big[A_i(L)\Big]\,(y_i - x_i(L))$$

$$z_i(\ell) = \nabla f_\ell\Big[A_i(\ell)\Big]W_{\ell+1}^T z_i(\ell+1) \quad \ell = 0, \ldots, L-1$$

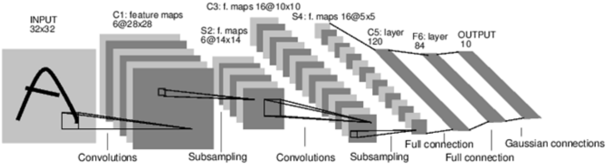
- $W \leftarrow W + \lambda \frac{\partial \mathcal{L}}{\partial W}$

Weight update

$$W_\ell \leftarrow W_\ell + \lambda \textstyle\sum_{i=1}^n z_i(\ell) x_i^T(\ell - 1)$$

Convolutional Neural Network (CNN)

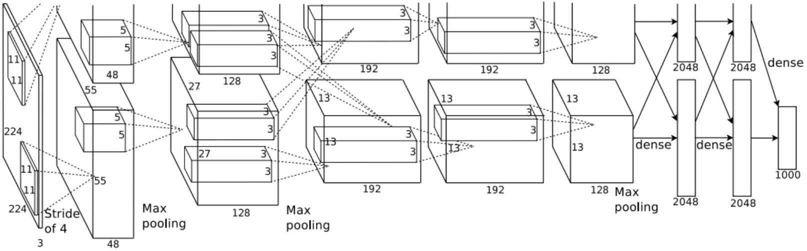
- Can be traced to *Neocognitron* of Kunihiko Fukushima (1979)
- Yann LeCun combined convolutional neural networks with back propagation (1989)
- Imposes **shift invariance** and **locality** on the weights
- Forward pass remains similar
- Backpropagation slightly changes – need to sum over the gradients from all spatial positions



Source: [LeCun et al., 1998]

AlexNet (2012)
Architecture

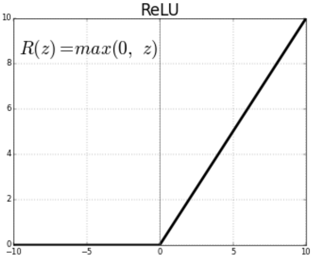
- 8 layers: first 5 convolutional, rest fully connected
- ReLU nonlinearity
- Local response normalization
- Max-pooling
- Dropout



Source: [Krizhevsky et al., 2012]

AlexNet (2012)
ReLU

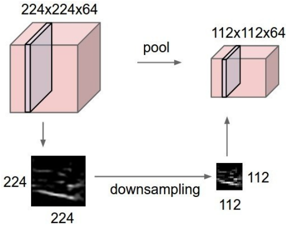
- Non-saturating function and therefore faster convergence when compared to other nonlinearities
- Problem of dying neurons

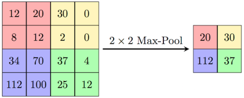


Source: `https://ml4a.github.io/ml4a/neural_networks/`

AlexNet (2012)
Max pooling

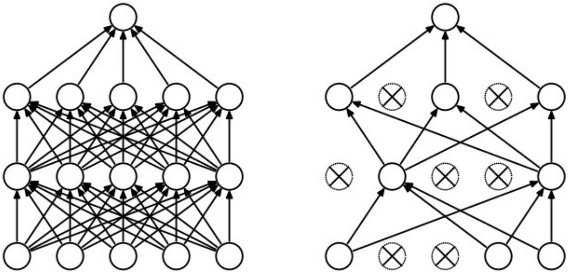
- Chooses maximal entry in every non-overlapping window of size 2×2, for example





Source: Stanford's CS231n github

AlexNet (2012)
Dropout



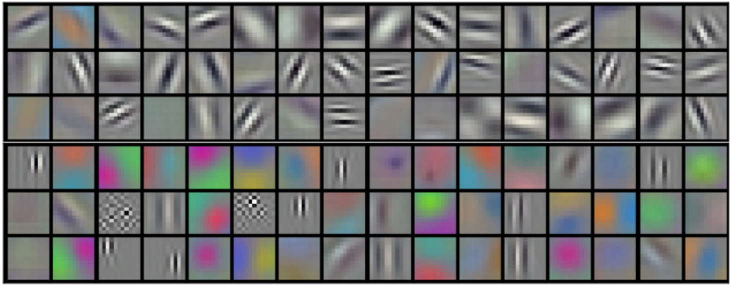
(a) Standard Neural Net (b) After applying dropout.

Source: [Srivastava et al., 2014]

- Zero every neuron with probability $1 - p$
- At test time, multiply every neuron by p

AlexNet (2012)
Training

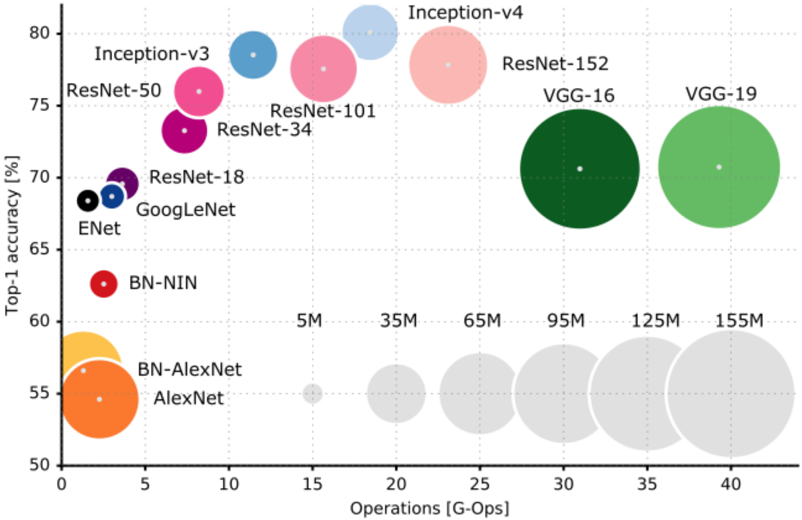
- Stochastic gradient descent
- Mini-batches
- Momentum
- Weight decay (ℓ_2 prior on the weights)



Filters trained in the first layer

Source: [Krizhevsky et al., 2012]

Characteristics of different networks



Source: Eugenio Culurciello

The need for regularization

- The number of training examples is 1.2 million
- The number of parameters is 5-155 million
- How does the network manage to generalize?

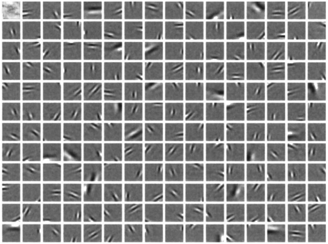
Implicit and explicit regularization

- Weight decay (ℓ_2 prior on the weights)
- ReLU soft non-negative thresholding operator. Implicit regularization of sparse feature maps
- Dropout – at test time, when no units dropped, gives sparser representations [Srivastava et. al 14']
- Dropout a particular form of ridge regression
- The structure of the network itself

Olshausen and Field (1996)

- Receptive fields in visual cortex are spatially localized, oriented and bandpass
- Coding natural images while promoting sparse solutions results in a set of filters satisfying these properties

$$\min_{\{\phi_i\}, a_i} \frac{1}{2} \left\| I - \sum_i \phi_i a_i \right\|_2^2 + \sum_i S(a_i), \tag{2}$$

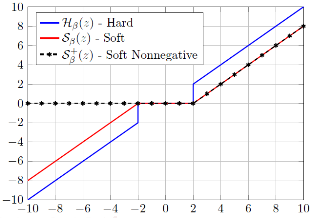


Trained filters ϕ_i

Source: [Olshausen and Field, 1996]

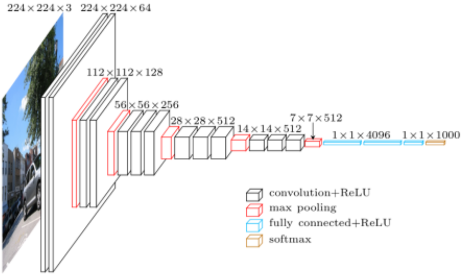
AlexNet vs. Olshausen and Field

- Why does AlexNet learn filters similar to Olshausen/Field?
- Is there an implicit sparsity-promotion in training network?
- How would classification results change if replace learned filters in first layer with analytically defined wavelets, e.g. Gabors?
- Filters in the first layer are spatially localized, oriented and bandpass. What properties do filters in remaining layers satisfy?
- Can we derive mathematically?



VGG (2014) [Simonyan and Zisserman, 2014]

- Deeper than AlexNet: 11-19 layers versus 8
- No local response normalization
- Number of filters multiplied by two every few layers
- Spatial extent of filters 3×3 in all layers
- Instead of 7×7 filters, use three layers of 3×3 filters
  - Gain intermediate nonlinearity
  - Impose a regularization on the 7×7 filters



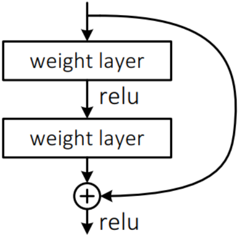
Source: `https://blog.heuritech.com/2016/02/29/`

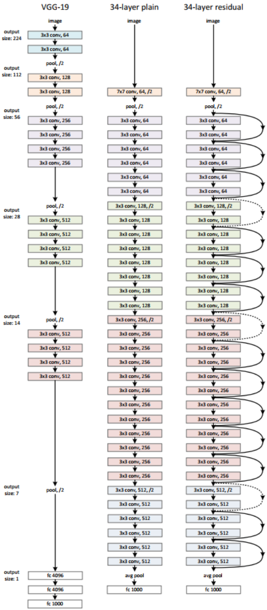
Optimization problems

- Formally, deeper networks contain shallower ones (i.e. consider no-op layers)
- **Observation:** Deeper networks not always lower training error
- **Conclusion:** Optimization process can't successfully infer no-op

ResNet (2015)

- Solves problem by adding skip connections
- Very deep: 152 layers
- No dropout
- Stride
- Batch normalization

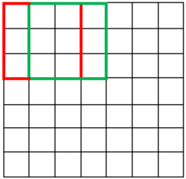




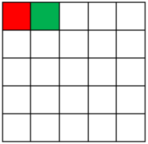
Source: Deep Residual Learning for Image Recognition

Stride

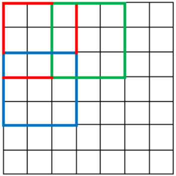
7 x 7 Input Volume



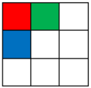
5 x 5 Output Volume



7 x 7 Input Volume



3 x 3 Output Volume



Source: `https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/`

Batch normalization

Algorithm 2 Batch normalization [Ioffe and Szegedy, 2015]

Input: Values of x over minibatch $x_1 \ldots x_B$, where x is a certain channel in a certain feature vector

Output: Normalized, scaled and shifted values $y_1 \ldots y_B$

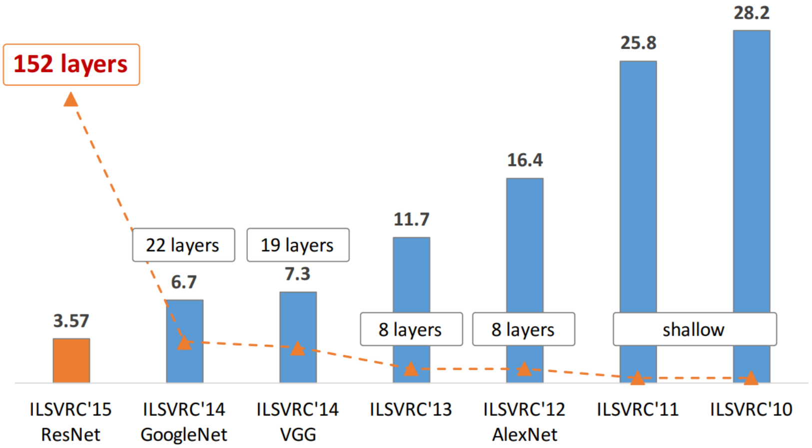
1: $\mu = \frac{1}{B} \sum_{b=1}^{B} x_b$
2: $\sigma^2 = \frac{1}{B} \sum_{b=1}^{B} (x_b - \mu)^2$
3: $\hat{x}_b = \frac{x_b - \mu}{\sqrt{\sigma^2 + \epsilon}}$
4: $y_b = \gamma \hat{x}_b + \beta$

- Accelerates training and makes initialization less sensitive
- Zero mean and unit variance feature vectors

ResNet versus standard architectures

- **Standard architectures:** increasingly abstract features at each layer
- **ResNet:** a group of successive layers iteratively refine an estimated representation [Klaus Greff et. al ’17]
- Could we formulate a cost function that is being minimized in these successive layers?
- What is the relation between this cost function and standard architectures?

Depth as function of year



[He et al., 2016]

The question of depth

- Besides increasing depth, one can increase *width* of each layer to improve performance

  [Zagoruyko and Komodakis 17']
- Is there a reason for increasing depth over width or vice versa?
- Is having many filters in same layer somehow detrimental?
- Is having many layers not beneficial after some point?

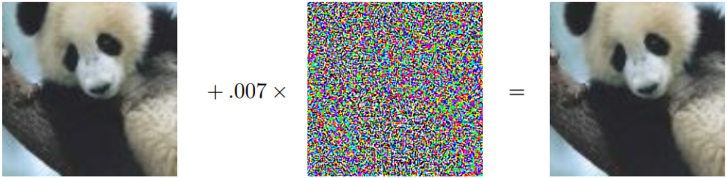
Linear separation

- Inputs are not linearly separable but their deepest representations are
- What happens during forward pass that makes linear separation possible?
- Is separation happening gradually with depth or abruptly at a certain point?

Transfer learning

- Filters learned in first layers of a network are transferable from one task to another
- When solving another problem, no need to retrain the lower layers, just fine tune upper ones
- Is this simply due to the large amount of images in ImageNet?
- Does solving many classification problems simultaneously result in features that are more easily transferable?
- Does this imply filters can be learned in unsupervised manner?
- Can we characterize filters mathematically?

Adversarial examples



$\boldsymbol{x}$
"panda"
57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"nematode"
8.2% confidence

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$
"gibbon"
99.3 % confidence

[Goodfellow et al., 2014]

- Small but malicious perturbations can result in severe misclassification
- Malicious examples generalize across different architectures
- What is source of instability?
- Can we robustify network?

Visualizing deep convolutional neural networks using natural pre-images

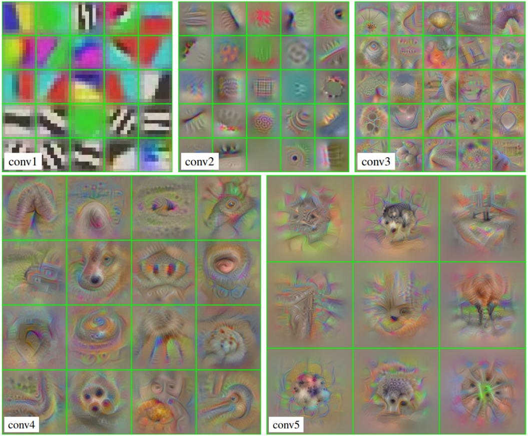
- Filters in first layer of CNN are easy to visualize, while deeper ones are harder
- *Activation maximization* seeks input image maximizing output of the i-th neuron in the network
- Objective

$$x^* = \arg\min_x \mathcal{R}(x) - \langle \Phi(x), e_i \rangle \tag{3}$$

- e_i is indicator vector
- $\mathcal{R}(x)$ is simple natural image prior

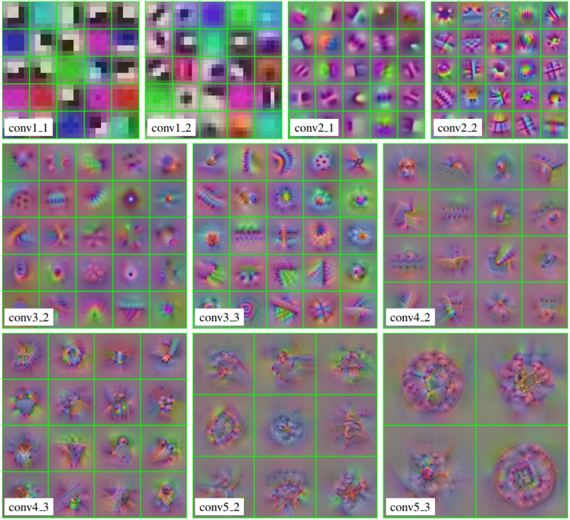
Visualizing VGG

- Gabor-like images in first layer
- More sophisticated structures in the rest



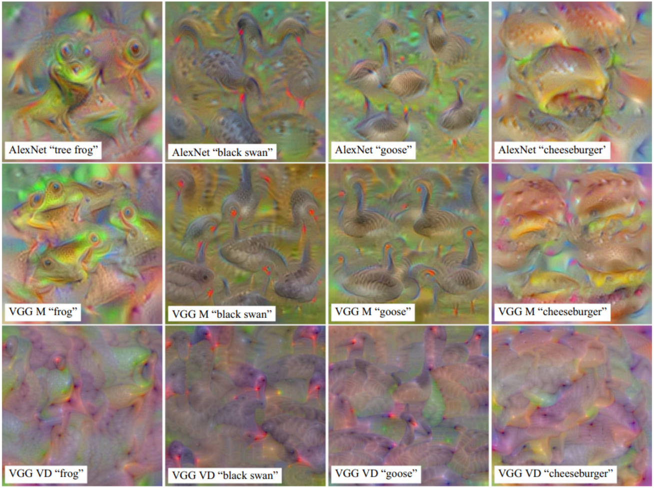
[Mahendran and Vedaldi, 2016]

Visualizing VGG VD



[Mahendran and Vedaldi, 2016]

Visualizing CNN

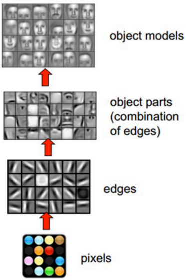


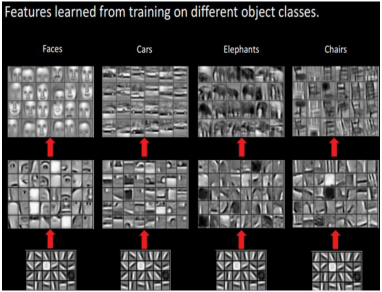
[Mahendran and Vedaldi, 2016]

Geometry of images

- Activation maximization seeks input image maximizing activation of certain neuron
- Could we span all images that excite a certain neuron?
- What geometrical structure would these images create?

Lecture 2, in overview





References I

Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Mahendran, A. and Vedaldi, A. (2016). Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120(3):233–255.

Olshausen, B. A. and Field, D. J. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.