CPSC 540: Machine Learning

Fully-Convolutional Networks, Empirical Bayes

Mark Schmidt

University of British Columbia

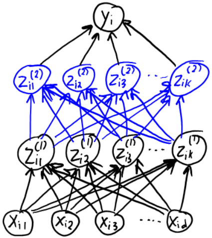
Winter 2017

Admin

- Assignment 4:
  - 2 late days to hand in tonight.
- Assignment 5 coming soon.
- Project description coming soon.
- Final details coming soon.
- Bonus lecture on April 10th (same time and place)?

Last Time: Deep Neural Networks

- In deep neural networks we use multiple layers of latent variables,



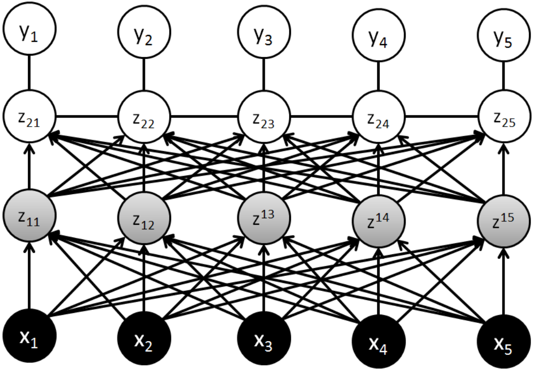
- Mathematically, with 3 hidden layers the classic model uses

$$y^i = w^T \underbrace{h(W^3 \underbrace{h(W^2 \underbrace{h(W^1 x^i)}_{z^{i1}})}_{z^{i2}})}_{z^{i3}}.$$

- We can think of this as a model that learns the features.
- The z^{im} are deterministic: less powerful than stochastic but inference is easy.

Last Time: Deep CRFs

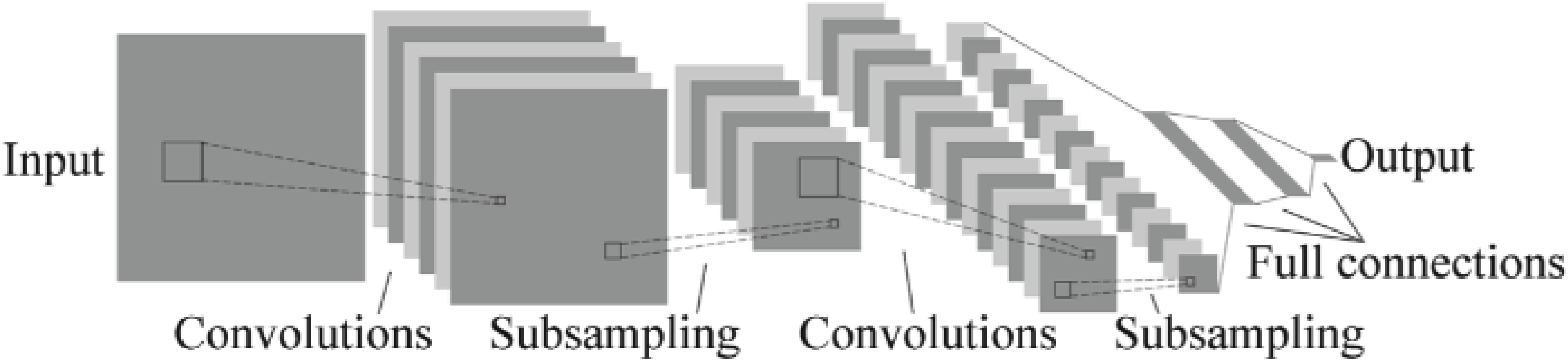
- We can combine neural networks with other models like CRFs and HMMs:



- Neural network models the features.
- Hidden Markov chain learns the "parts" and their dependence.
- CRF lets us condition on x so inference is easy.

Outline

Last Time: Convolutional Neural Networks



- Convolutional neural networks classically have 3 layer "types":
  - Fully connected layer: usual neural network layer with unrestricted W.
  - Convolutional layer: restrict W to results of several convolutions.
  - Pooling layer: downsamples result of convolution.

http://scs.ryerson.ca/~aharley/vis/harley_vis_isvc15.pdf

AlexNet Convolutional Neural Network

- ImageNet 2012 won by AlexNet:
  - 15.4% error vs. 26.2% for closest competitor.
  - 5 convolutional layers.
  - 3 fully-connected layers.
  - SG with momentum.
  - ReLU non-linear functions.
  - Data translation/reflection/ cropping.
  - L2-regularization + Dropout.
  - 5-6 days on two GPUs.

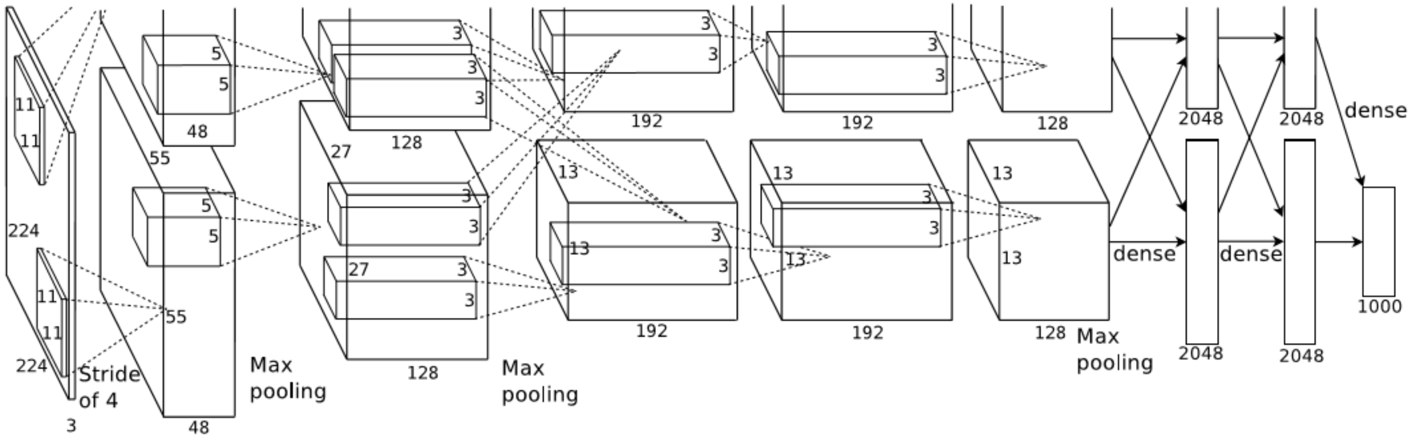


Figure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities between the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and the number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–4096–4096–1000.

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

AlexNet Convolutional Neural Network

- ImageNet 2012 won by AlexNet:
  - 15.4% error vs. 26.2% for closest competitor.

Gaussian times sine/cosine: "Gabor" filters

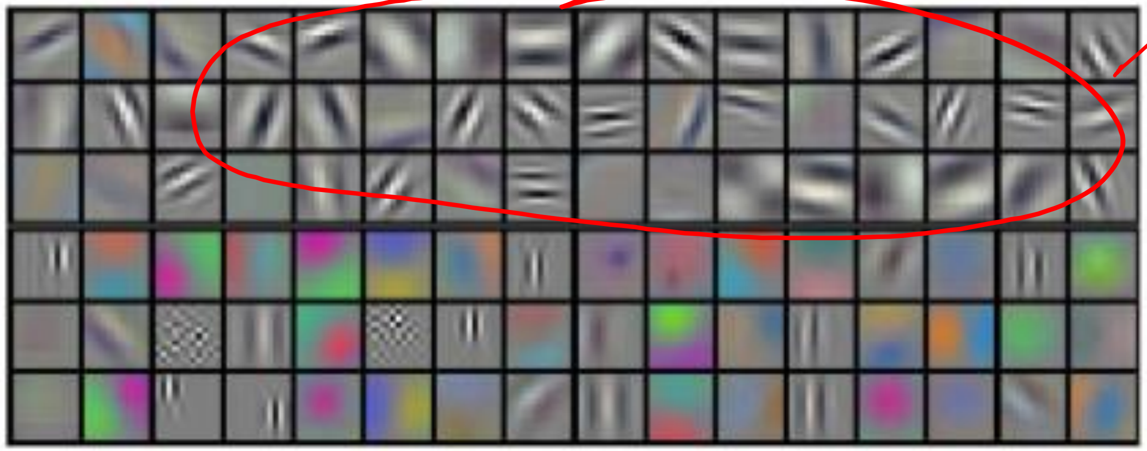
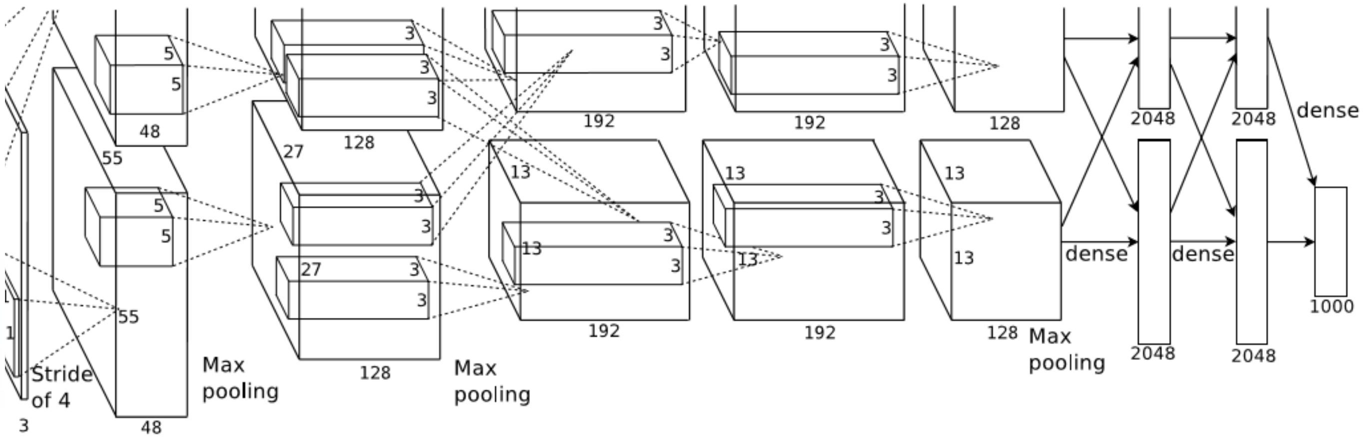


Figure 3: 96 convolutional kernels of size 11×11×3 learned by the first convolutional layer on the 224×224×3 input images. The



ure 2: An illustration of the architecture of our CNN, explicitly showing the delineation of responsibilities veen the two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts ne bottom. The GPUs communicate only at certain layers. The network's input is 150,528-dimensional, and number of neurons in the network's remaining layers is given by 253,440–186,624–64,896–64,896–43,264–6–4096–1000.

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

ZFNet Convolutional Neural Network

- ImageNet 2013 won by variation of AlexNet called ZF Net:
  - 11.2% error (now using 7x7 instead of 11x11).
  - Introduced deconvolutional networks to visualize what CNNs learn.

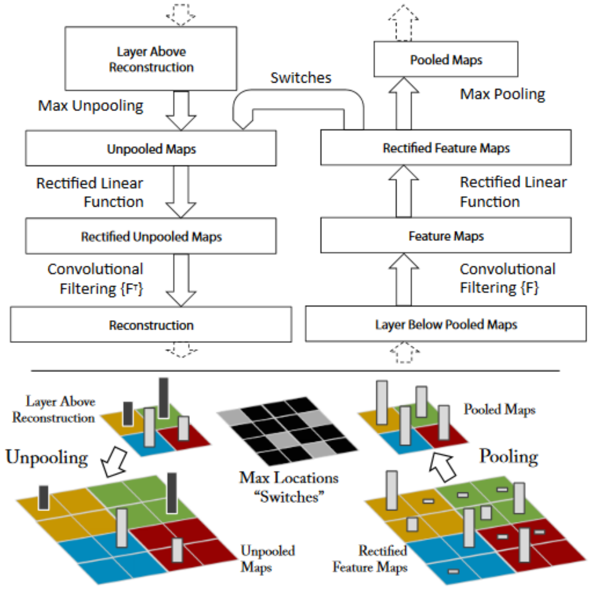
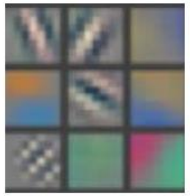
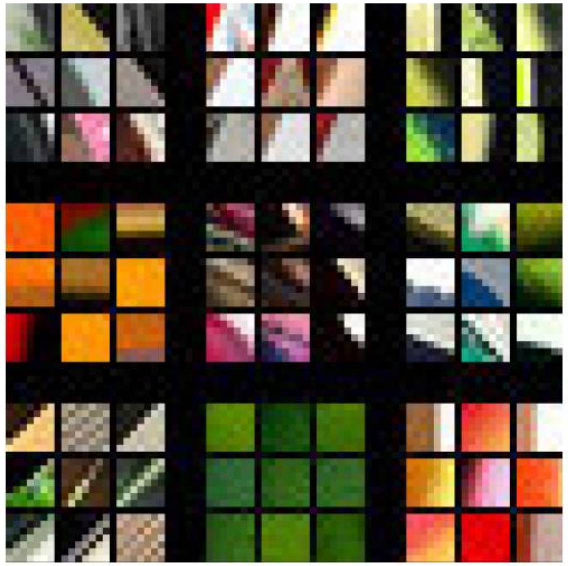


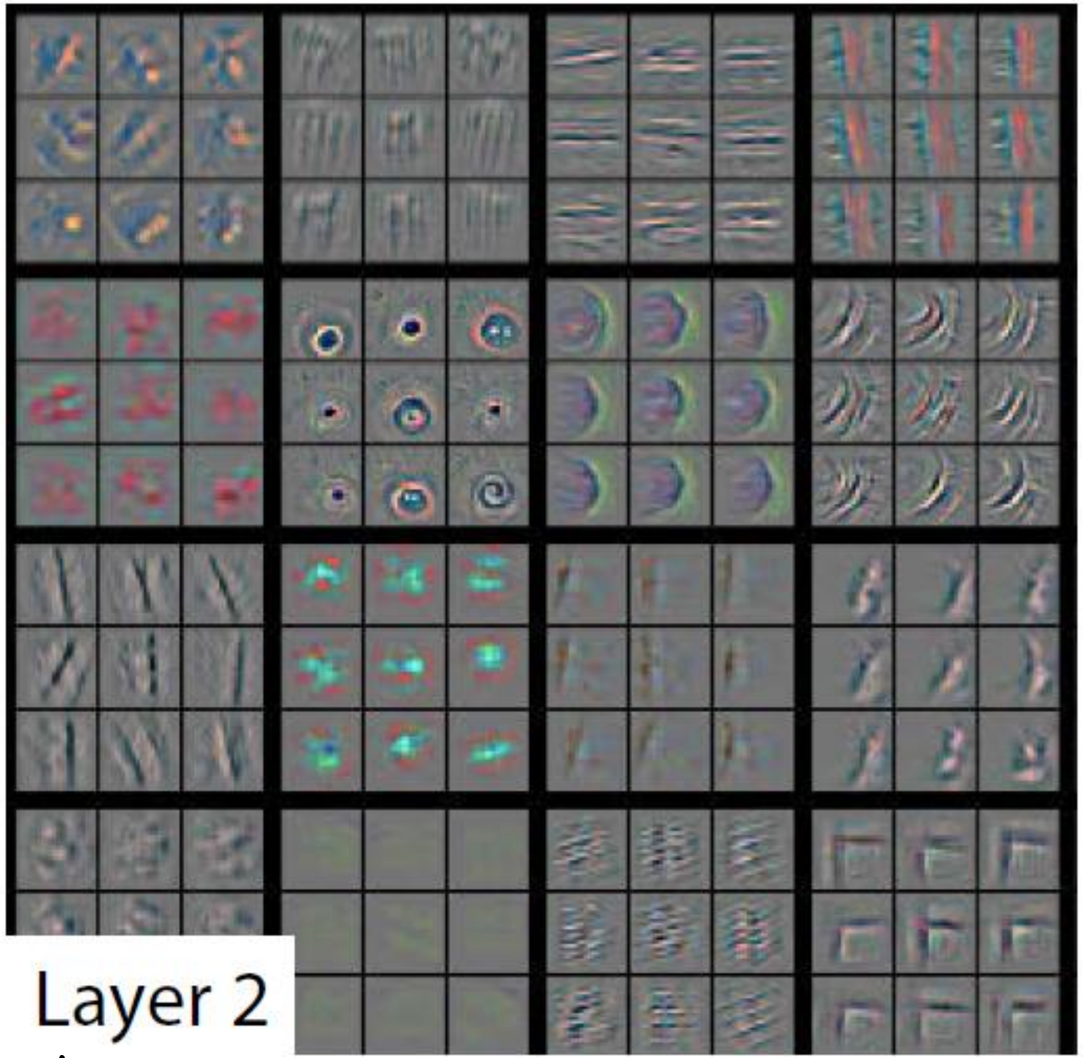
Figure 1. Top: A deconvnet layer (left) attached to a convnet layer (right). The deconvnet will reconstruct an approximate version of the convnet features from the layer beneath. Bottom: An illustration of the unpooling operation in the deconvnet, using *switches* which record the location of the local max in each pooling region (colored zones) during pooling in the convnet.

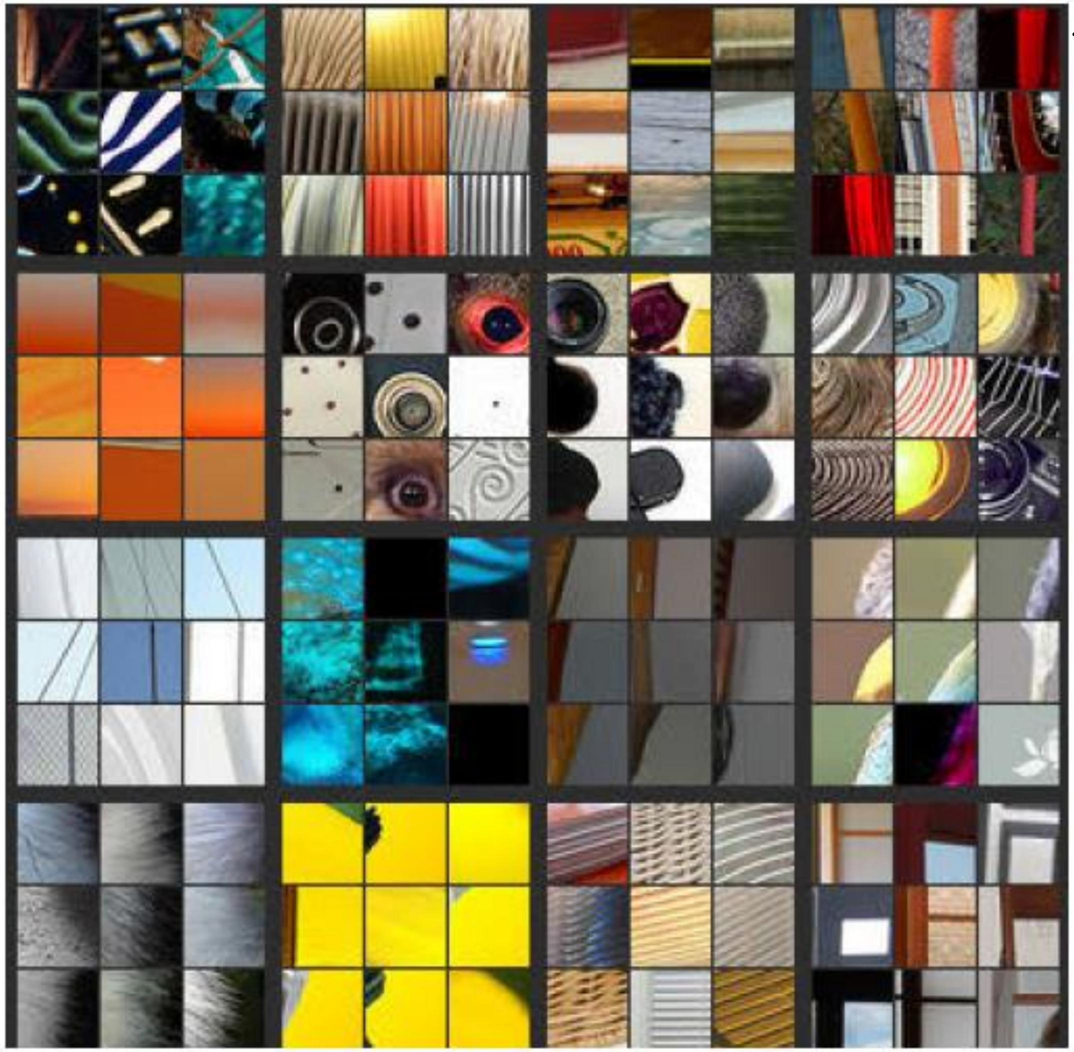


Layer 1



ZFNet Convolutional Neural Network



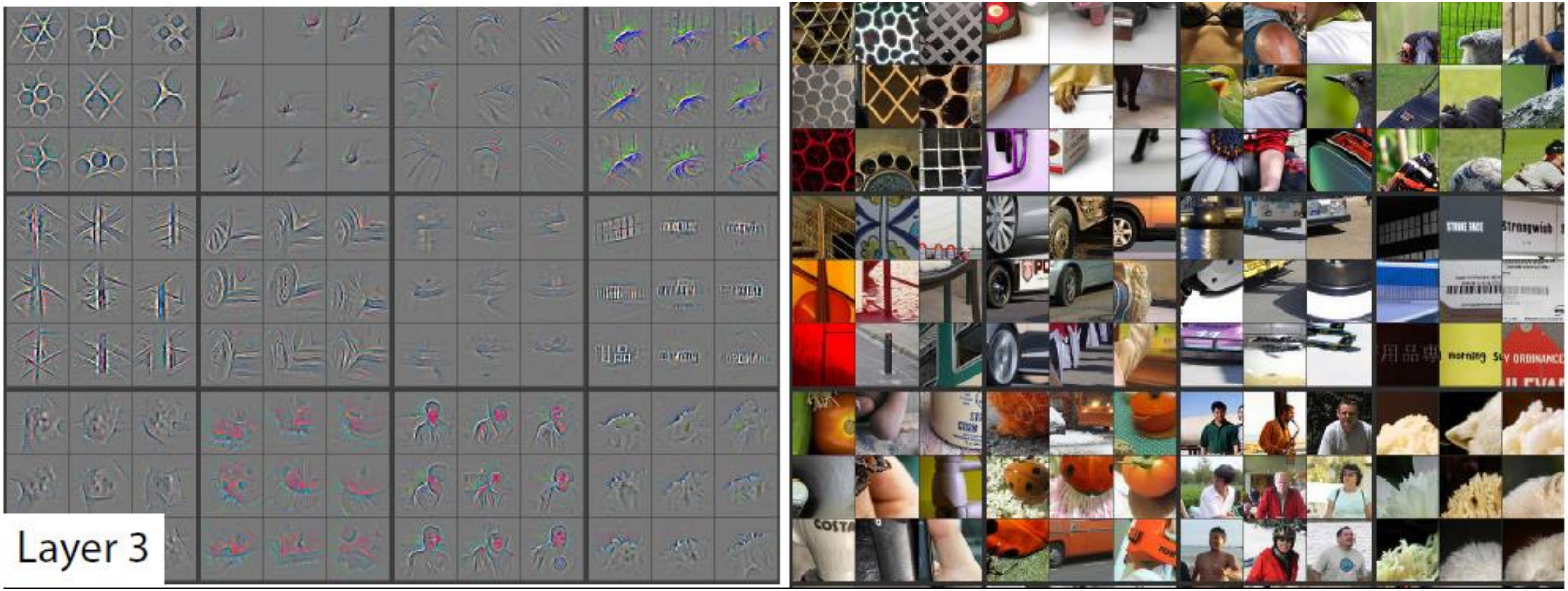


Patch from data giving largest response

https://arxiv.org/pdf/1311.2901v3.pdf

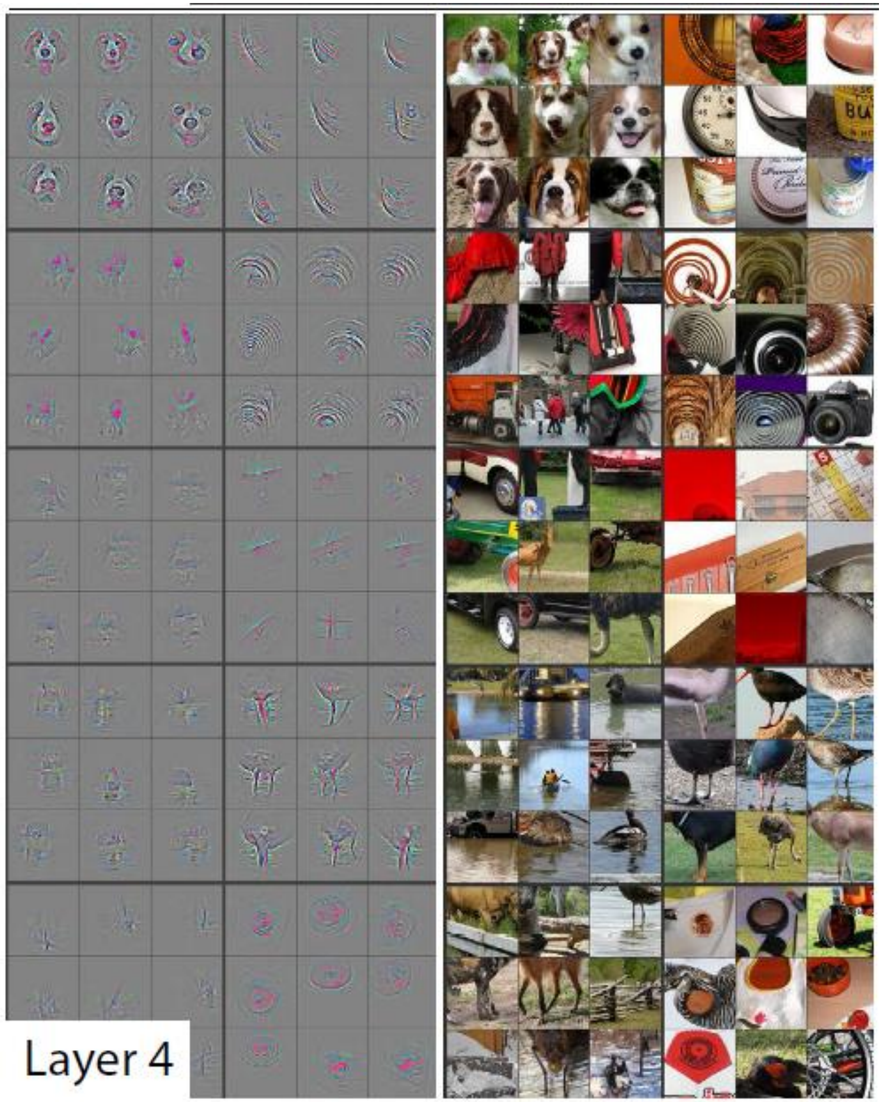
Deconvolution network giving patch that leads to largest response

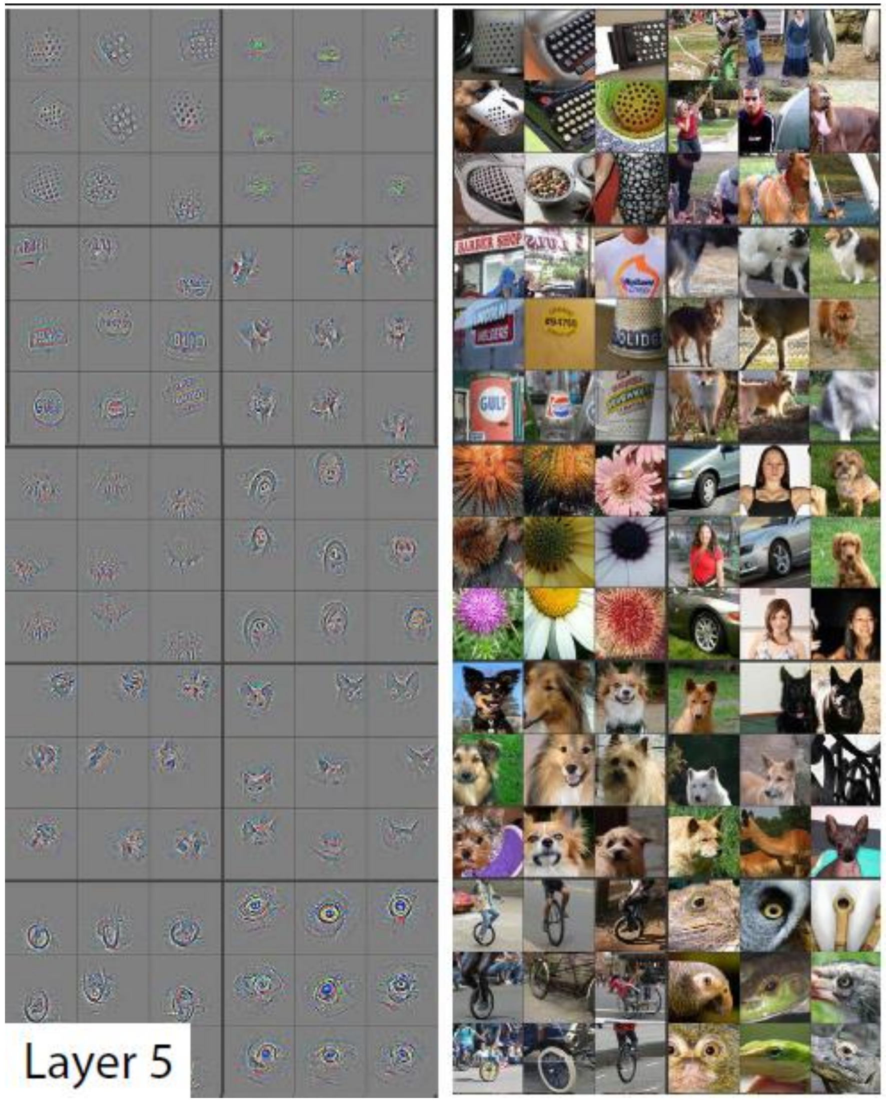
ZFNet Convolutional Neural Network



https://arxiv.org/pdf/1311.2901v3.pdf

ZFNet Convolutional Neural Network

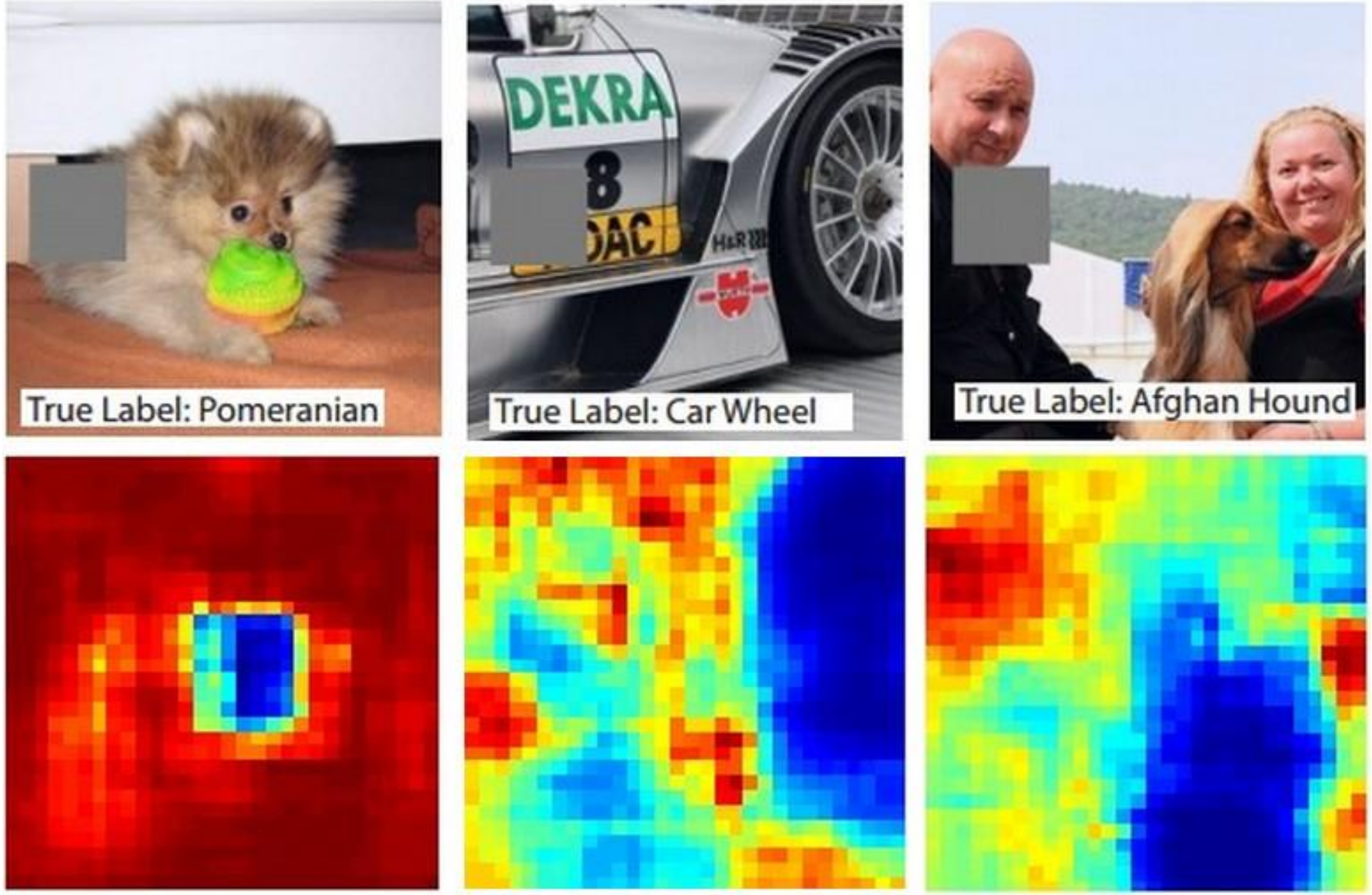




https://arxiv.org/pdf/1311.2901v3.pdf

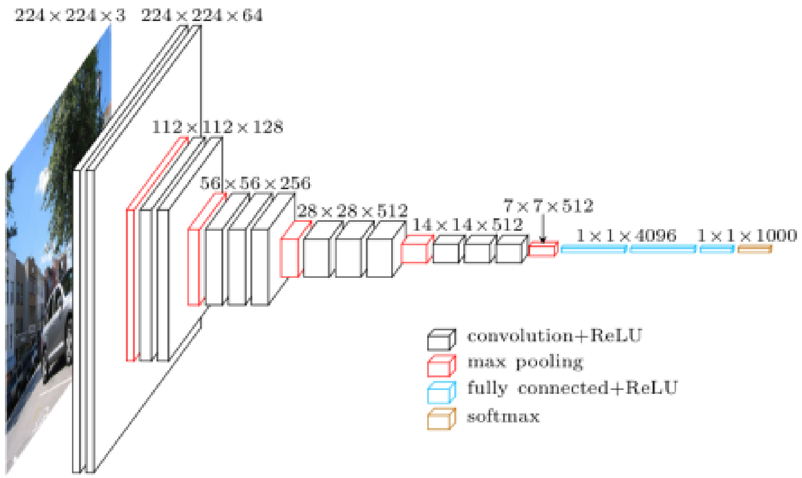
ZFNet Convolutional Neural Network

- Looked at how prediction changes if we hide part of the image:



VGG Convolutional Neural Network

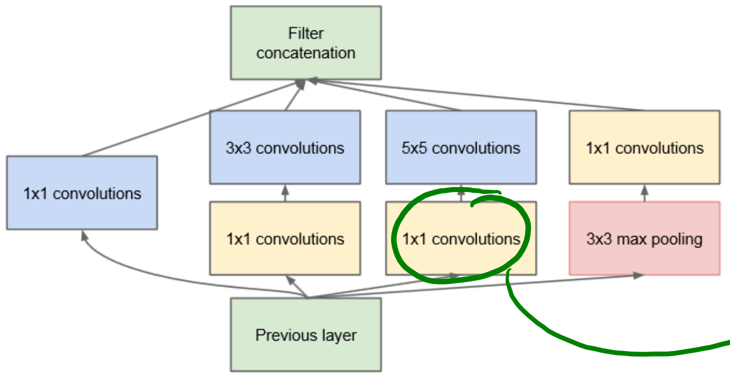
- Image 2014 "Localization" Task won by a 19-layer VGG network:
  - 7.3% error for classification (2nd place).
  - Uses 3x3 convolution layers with stride of 1:
    - 3x3 followed by 3x3 simulates a 5x5, and another 3x3 simulates a 7x7, and so on.
    - Speeds things up and reduces number of parameters.
    - Increases number of non-linear ReLU operations.
  - "Deep and simple": variants of VGG are the most popular CNNs.



https://www.cs.toronto.edu/~frossard/post/vgg16/

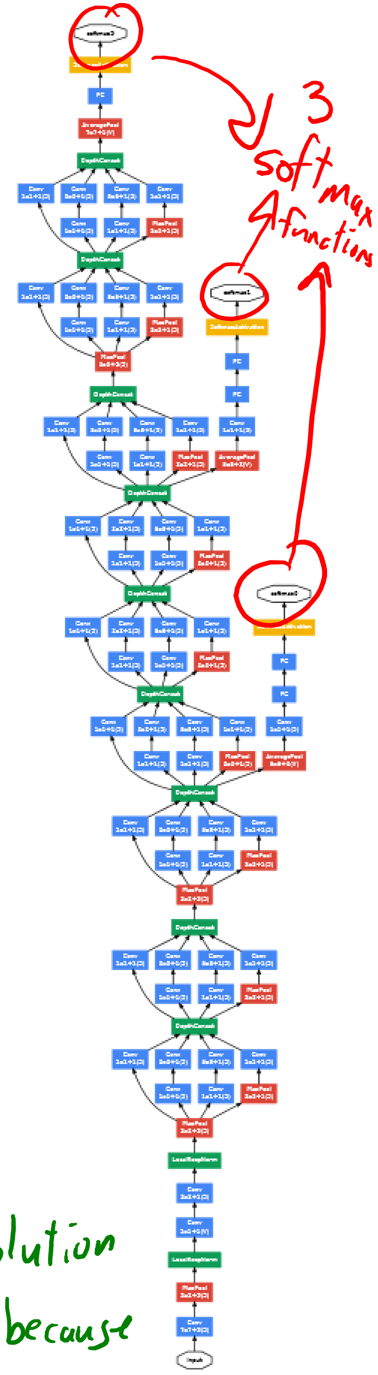
GoogLeNet

- Image 2014 classification task won by GoogLeNet:
  - 6.7% errors.
  - 22 layers
    - No fully connected layers.
    - During training, try to predict label at multiple locations.
      - During testing, just take the deepest predictions.
    - "Inception" modules: combine convolutions of different sizes.



(b) Inception module with dimensionality reduction

"1x1" convolution makes sense because these are first 2 dimensions of 3D conv.



https://www.cs.unc.edu/~wliu/papers/GoogLeNet.pdf
http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

ResNet

- Image 2015 won by Resnet (all 5 tasks):
  - 3.6% error (below estimated 5% human error).
  - 152 layers (2-3 weeks on 8 GPUs to train).
  - "Residual learning" allows better performance with deep networks:
    - Include input to layer in addition to non-linear transform.

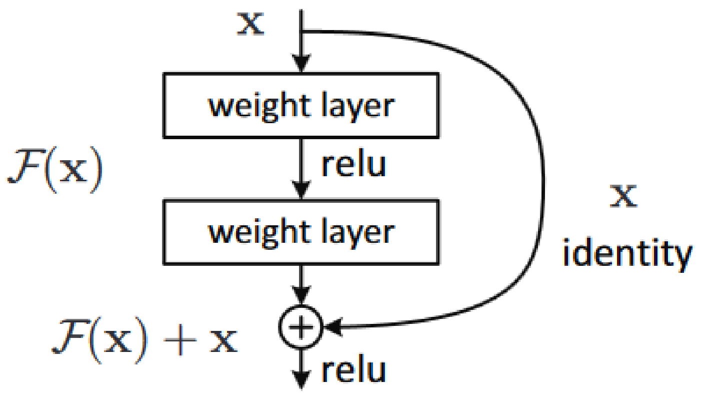


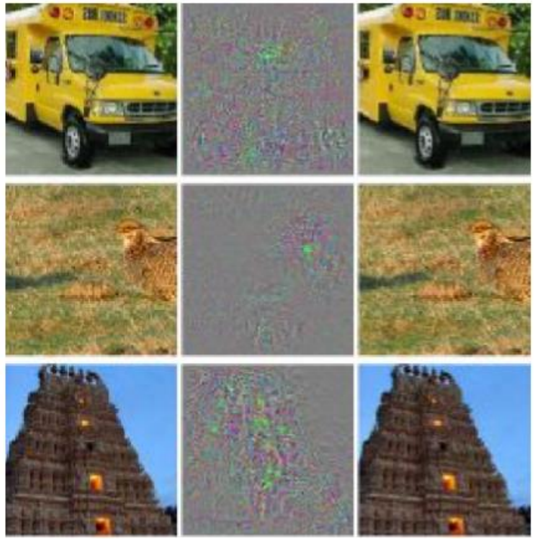
Figure 2. Residual learning: a building block.

    - Network just focuses on "residual": what is not captured in the input signal.

https://arxiv.org/pdf/1512.03385v1.pdf

Mission Accomplished?

- For speech recognition and object detection:
  - No other methods have ever given the current level of performance.
  - But, we also don't know how to scale up other universal approximators.
  - There is likely some overfitting to these particular tasks.
- Despite high-level of abstraction, deep CNNs are easily fooled:
  - But progress on fixing 'blind spots'.



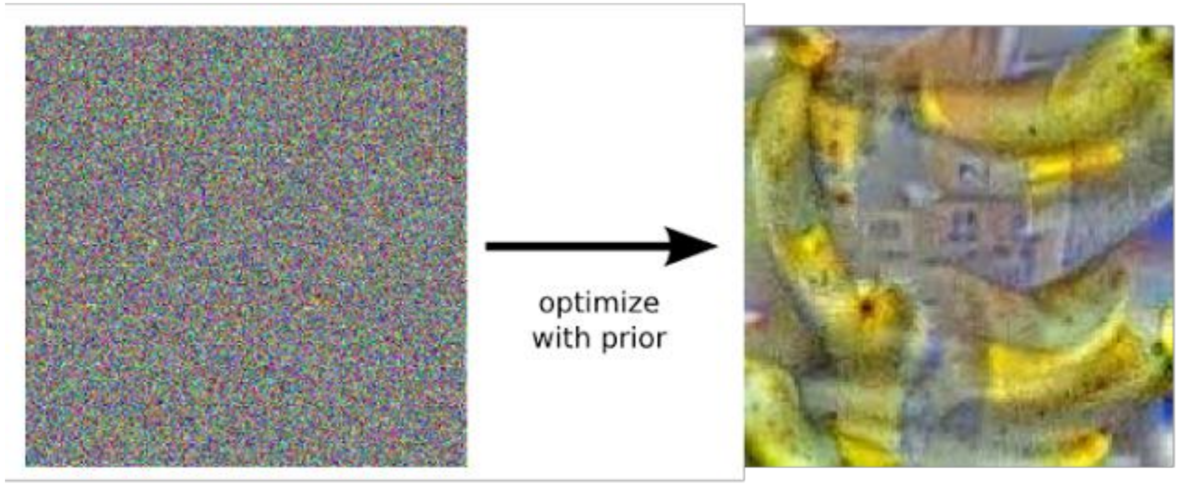
- Do we really need 1000 training images for every object?
  - Active research topic.

http://cs.nyu.edu/~zaremba/docs/understanding.pdf

Inceptionism

- A crazy idea:
  - Instead of weights, use backpropagation to take gradient with respect to x.
  - Available using the same message-passing algorithm.
- Inceptionism with trained network:
  - Fix the label y (e.g., “banana”).
  - Start with random noise image x.
  - Use gradient descent on image x.
  - Add total variation regularization:
    - Encourages spatial smoothness.
    - Equivalent to decoding in a UGM.

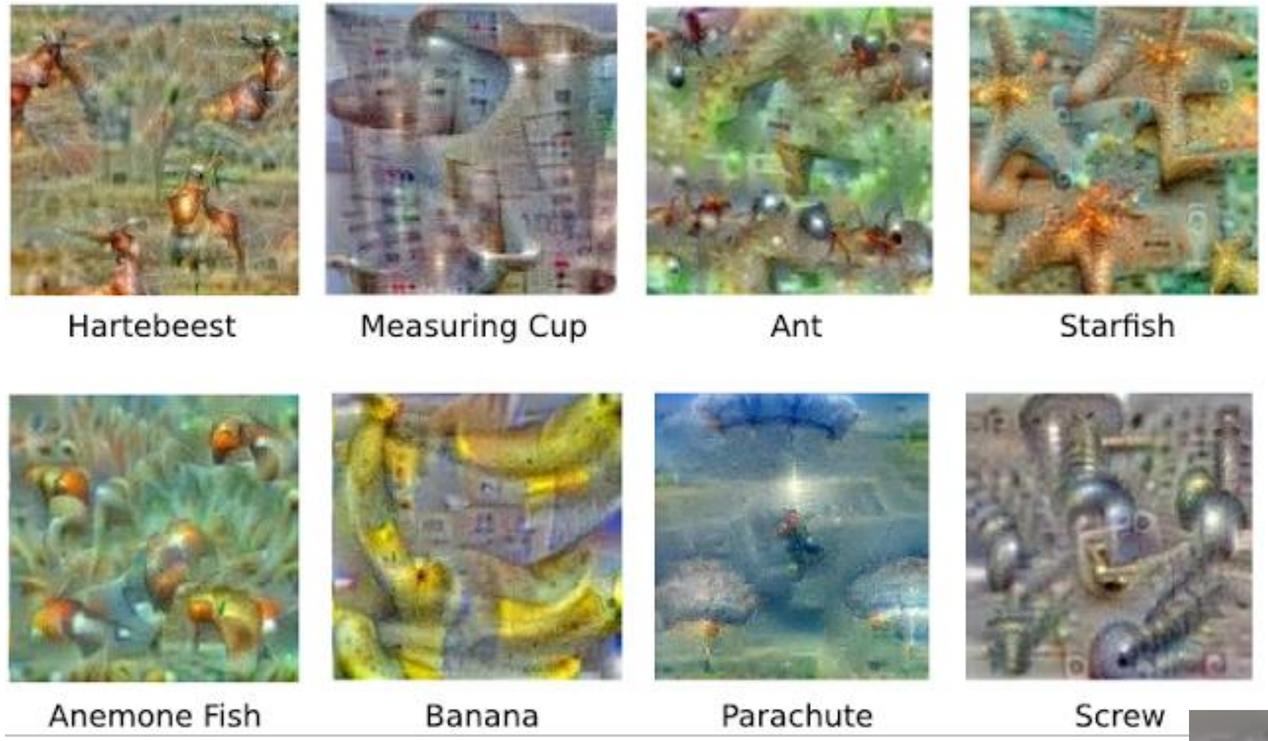
"Show what you think a banana looks like."



http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html

Inceptionism

- Inceptionism for different class labels:

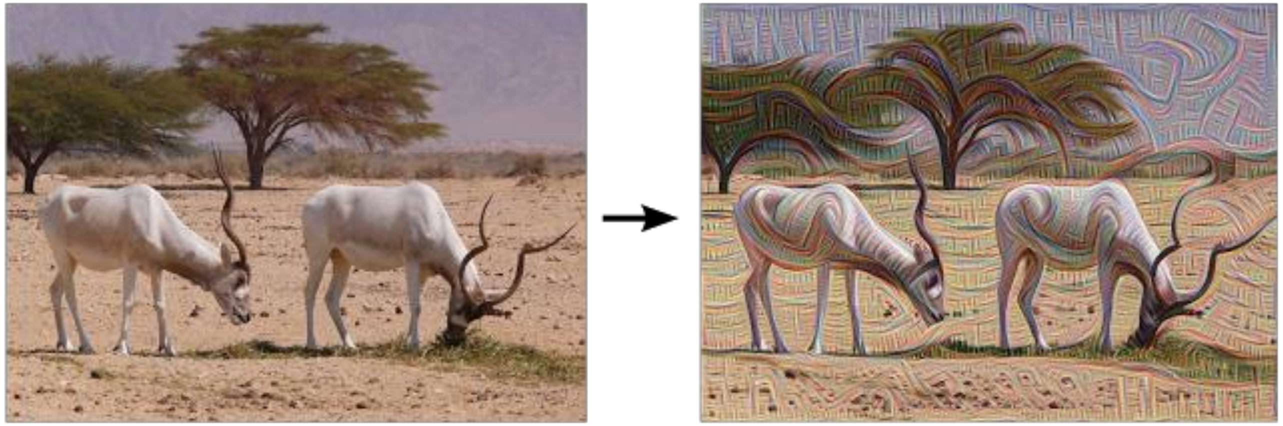


Dumbbell

http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html

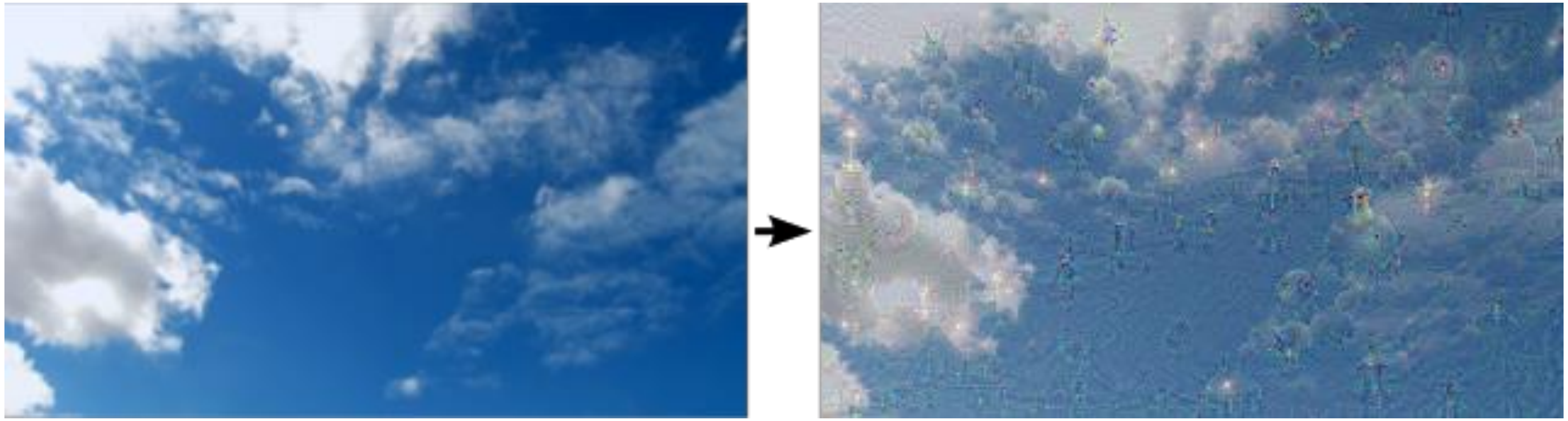
Inceptionism

- Inceptionism where we try to match z_c^m values instead of y.
  - Shallow ‘m’:



Inceptionism

- Inceptionism where we try to match z_c^m values instead of y.
  - Deepest ‘m’:



"Admiral Dog!"

"The Pig-Snail"

"The Camel-Bird"

"The Dog-Fish"

http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html

Inceptionism

- Inceptionism where we try to match z_c^m values instead of y.
  - "Deep dream" starts from random noise:

  - Inceptionism gallery
  - Deep Dream video

http://googleresearch.blogspot.ca/2015/06/inceptionism-going-deeper-into-neural.html

Artistic Style Transfer

- Artistic style transfer:
  - Given a content image ‘C’ and a style image ‘S’.
  - Make a image that has content of ‘C’ and style of ‘S’.

Content:

Style:

https://commons.wikimedia.org/wiki/File:Tuebingen_Neckarfront.jpg
https://en.wikipedia.org/wiki/The_Starry_Night

Artistic Style Transfer

- Artistic style transfer:
  - Given a content image 'C' and a style image 'S'.
  - Make a image that has content of 'C' and style of 'S'.

- CNN-based approach applies gradient descent with 2 terms:
  - Loss function: match deep latent representation of content image 'C':
    - Difference between z_c^m for deepest 'm' between x and 'C'.
  - Regularizer: match all latent representation covariances of style image 'S'.
    - Difference between covariance of z_c^m for all 'm' between x and 'S'.

Artistic Style Transfer

http://arxiv.org/pdf/1508.06576v2.pdf

Image Gallery

Examples

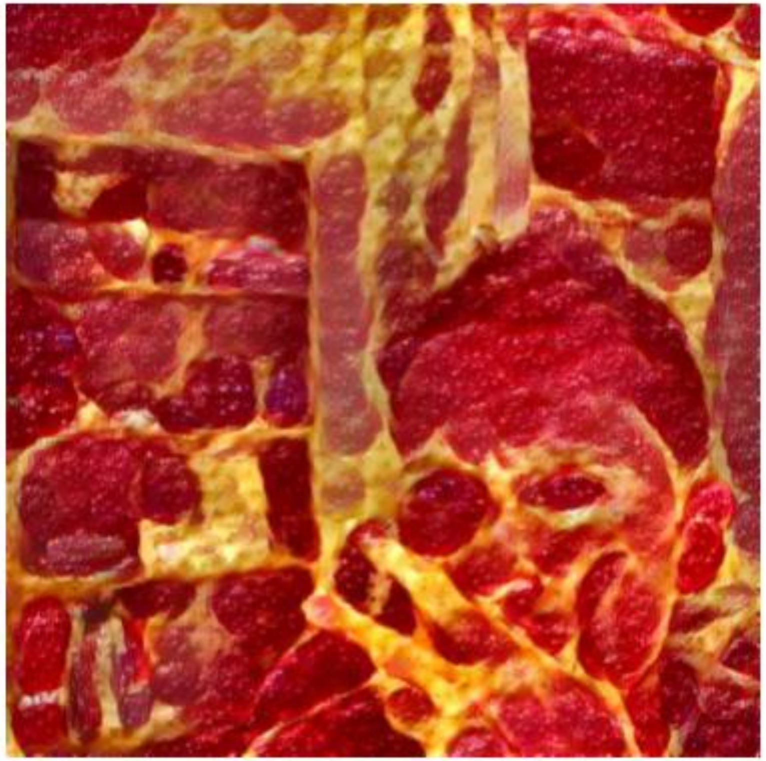


Figure: **Left:** My friend Grant, **Right:** Grant as a pizza

Artistic Style Transfer

- Recent methods:
  - Find ‘x’ that matches image patches ‘z’ space apply TV-regularization.

Input A   Input B   Content A + Style B   Content B + Style A

https://arxiv.org/pdf/1601.04589.pdf

Artistic Style Transfer

- Recent methods:
  - Find ‘x’ that matches image patches ‘z’ space apply TV-regularization.

Input style Input content Ours

https://arxiv.org/pdf/1601.04589.pdf

Artistic Style Transfer for Video

- Combining style transfer with optical flow:
  - https://www.youtube.com/watch?v=Khuj4ASldmU
- Videos from Ricky's paper:

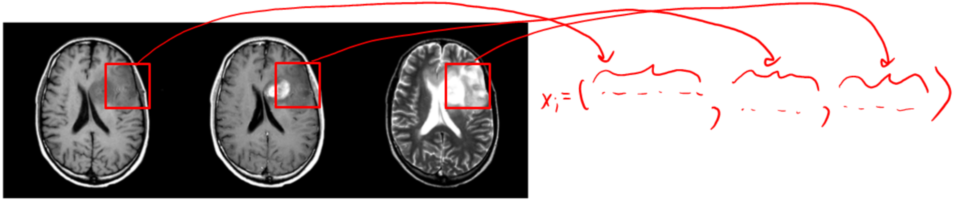

Outline

Motivation: Beyond Classification

- Convolutional structure simplifies the learning task:
  - Parameter tieing means we have more data to estimate each parameter.
  - Sparsity drastically reduces number of parameters.
- But many vision tasks are not image classification tasks.
  - Pixel labeling ("tumour" or "not tumour").
  - Depth estimation.
  - Pose estimation.
  - Optical flow.
  - Uncovering 3D geometry.

Straightforward CNN Extensions to Pixels Labeling

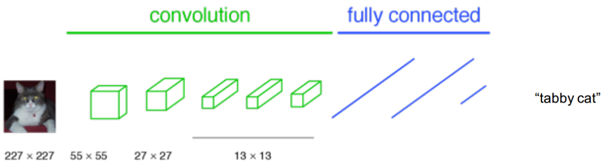
- Approach 1: apply an existing CNN to classify pixel given neighbourhood.
  - Misses long range dependencies in the image.
  - It's slow: for 200 by 200 image, need to do forward propagation 40000 times.



- Approach 2: add per-pixel labels to final layer of an existing CNN.
  - Fully-connected layers lose spatial information.
  - Relies on having fixed-size images.

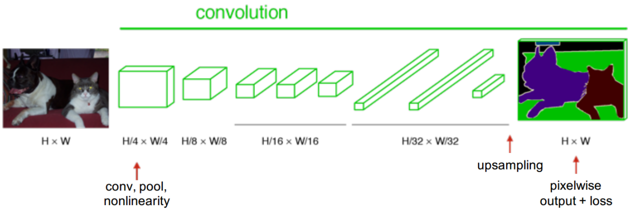
Fully-Convolutional Neural Networks

- Fully-convolutional neural networks (FCNs): CNNs with no fully-connected layers.
  - All layers maintain spatial information.



Fully-Convolutional Neural Networks

- Fully-convolutional neural networks (FCNs): CNNs with no fully-connected layers.
  - All layers maintain spatial information.

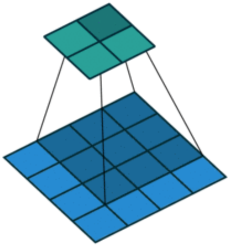
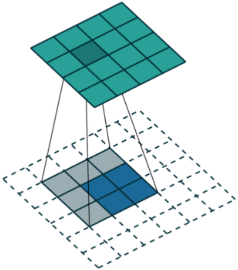


https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/image_segmentation.html

- Final layer upsamples to original image size.
  - With a learned “transposed convolution”.

Transposed Convolution Layer

- The upsampling layer is also called a transposed convolution or deconvolution.
  - Implemented as another convolution.

Convolution:  Transposed: 

https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/image_segmentation.html

- Reasons for the names:
  - "Tranposed" because sparsity pattern is transpose of a downsampling convolution.
  - "Deconvolution" is not related to the "deconvolution" in signal processing.

Fully-Convolutional Neural Networks

- FCNs are getting state of the art results on many tasks.

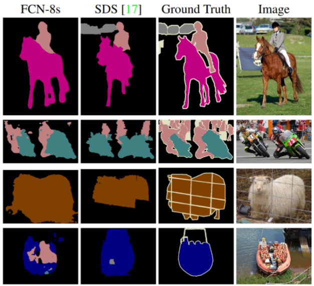


Figure 6. Fully convolutional segmentation nets produce state-of-the-art performance on PASCAL. The left column shows the output of our highest performing net, FCN-8s. The second shows the segmentations produced by the previous state-of-the-art system

https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf

- Buzzword is computer vision is that this is an end-to-end solution.
  - No super-pixels, object proposals, merging results from multiple classifiers, and so on.

Variationas on FCNs

- The transposed convolution at the last layer can lose a lot of resolution.
- One option is to adding "skip" connections from earlier higher-resolution layers.

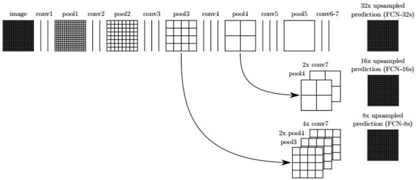
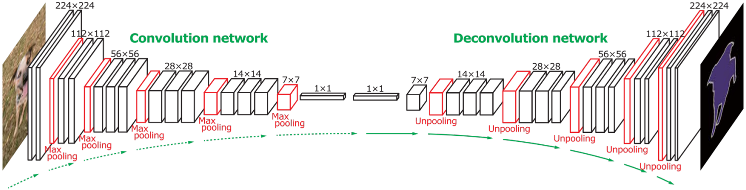


Figure 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Pooling and prediction layers are shown as grids that reveal relative spatial coarseness, while intermediate layers are shown as vertical lines. First row (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Second row (FCN-16s): Combining predictions from both the final layer and the pool4 layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Third row (FCN-8s): Additional predictions from pool3, at stride 8, provide further precision.

https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcn.pdf

- Another framework addressing this is deconvolutional networks:



Combining FCNs and CRFs

- Another way to address this is combining FCNs and CRFs.

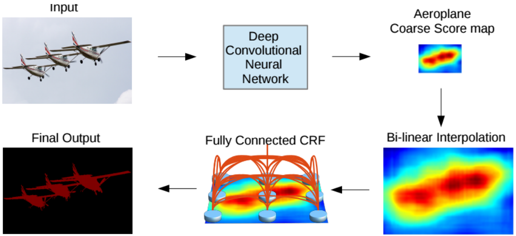


Fig. 1: Model Illustration. A Deep Convolutional Neural Network such as VGG-16 or ResNet-101 is employed in a fully convolutional fashion, using atrous convolution to reduce the degree of signal downsampling (from 32x down 8x). A bilinear interpolation stage enlarges the feature maps to the original image resolution. A fully connected CRF is then applied to refine the segmentation result and better capture the object boundaries.

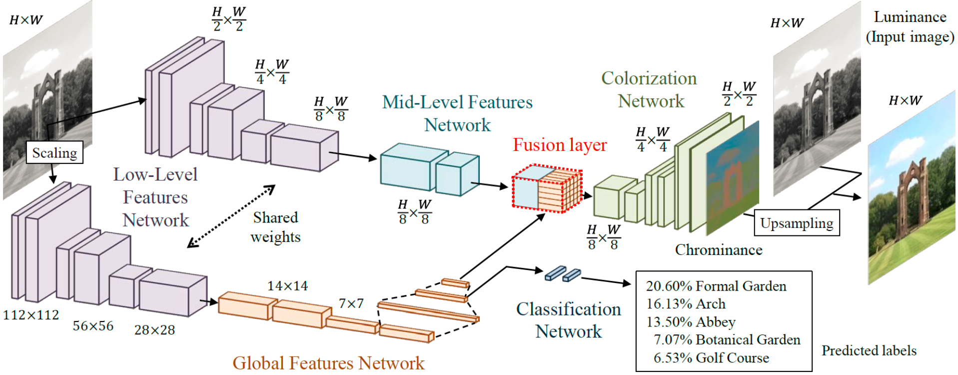
https://arxiv.org/pdf/1606.00915.pdf

- DeepLab uses a fully-connected pairiwse CRF on output layer.
  - Uses an efficient algorithm for mean-field with Gaussian potentials.

Image Colourization

- There now exist variations for all the standard vision tasks.
    - Depth estimation, pose estimation, optical flow, 3D geometry, and so on.
    - A bit hard to track the progress at the moment.

- Image colorization network:



http://hi.cs.waseda.ac.jp/~iizuka/projects/colorization/en

Image Colourization

- Image colorization results:

Colorado National Park, 1941 Textile Mill, June 1937 Berry Field, June 1909 Hamilton, 1936

http://hi.cs.waseda.ac.jp/~iizuka/projects/colorization/en

- Gallery:
  http://hi.cs.waseda.ac.jp/~iizuka/projects/colorization/extra.html
- Video: https://www.youtube.com/watch?v=ys5nMO4Q0iY

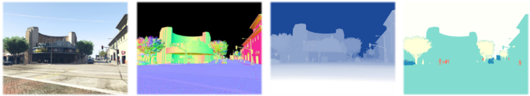
Where does data come from?

- Unfortunately, getting densely-labeled is often hard.

- For pixel labeling and depth estimation, we explored getting data from GTA V:

Video game

Google street view



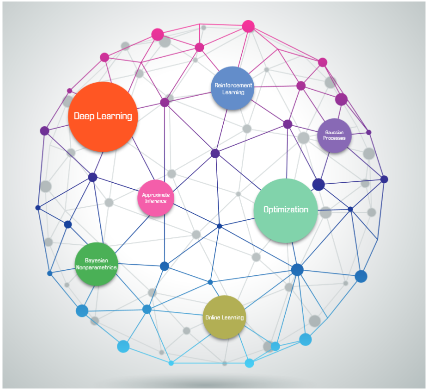
- Recent works use that you don't need full labeling.
    - Unobserved children in DAG don't induce dependencies.

Outline

Motivation: Controlling Complexity

- For many of these tasks, we need very complicated models.
  - We require multiple forms of regularization to prevent overfitting.
- In 340 we saw two ways to reduce complexity of a model:
  - Model averaging (ensemble methods).
  - Regularization (linear models).
- Bayesian methods combine both of these.
  - Average over models, weighted by posterior (which includes regularizer).

Current Hot Topics in Machine Learning



Bayesian learning includes:

- Gaussian processes.
- Approximate inference.
- Bayesian nonparametrics.

Why Bayesian Learning?

- Standard L2-regularized logistic regression steup:
  - Given finite dataset containing IID samples.
    - E.g., samples (x^i, y^i) with $x^i \in \mathbb{R}^d$ and $y^i \in \{-1, 1\}$.
  - Find "best" w by minimizing NLL with a regularizer to "prevent overfitting".

$$\hat{w} = \underset{w}{\text{argmin}} - \sum_{i=1}^{n} \log p(y^i | x^i, w) + \frac{\lambda}{2}\|w\|^2.$$

  - Predict labels of *new* example $\hat{x}$ using single weights w,

$$\hat{y} = \text{sgn}(\hat{w}^T \hat{x}).$$

- But data was random, so weights $\hat{w}$ are random variables.
  - This might put our trust in a w where posterior $p(w | X, y)$ is tiny.
- Bayesian approach: predictions based on rules of probability.

Problems with MAP Estimation

- Does MAP make the right decision?
  - Consider three hypothesese $\mathcal{H} = \{\text{“lands''}, \text{“crashes''}, \text{“explodes''}\}$ with posteriors:

$$p(\text{“lands''}|D) = 0.4, \quad p(\text{“crashes''}|D) = 0.3, \quad p(\text{“explodes''}|D) = 0.3.$$

  - The MAP estimate is “plane lands”, with posterior probability 0.4.
    - But probability of dying is 0.6.
    - If we want to live, MAP estimate doesn't give us what we should do.
- Bayesian approach averages models: says don't take plane.
- Bayesian decision theory: accounts for costs of different errors.

MAP vs. Bayes

- MAP (regularized optimization) approach maximizes over w:

$$\hat{w} \in \underset{w}{\operatorname{argmax}}\, p(w|X,y)$$
$$\equiv \underset{w}{\operatorname{argmax}}\, p(y|X,w)p(w) \qquad \text{(Bayes' rule, } w \perp X\text{)}$$
$$\hat{y} \in \underset{y}{\operatorname{argmax}}\, p(y|\hat{x},\hat{w}).$$

- Bayesian approach predicts by integrating over possible w:

$$p(\hat{y}|\hat{x},X,y) = \int_w p(\hat{y},w|\hat{x},X,y)dw \qquad \text{marginalization rule}$$
$$= \int_w p(\hat{y}|w,\hat{x},X,y)p(w|\hat{x},X,y)dw \qquad \text{product rule}$$
$$= \int_w p(\hat{y}|w,\hat{x})p(w|X,y)dw \qquad \hat{y} \perp X,y|\hat{x},w$$

- Considers all possible w, and weights prediction by posterior for w.

Motivation for Bayesian Learning

- Motivation for studying Bayesian learning:
  1. Optimal decisions using rules of probability and error costs.
  2. Gives estimates of variability/confidence.
     - E.g., this gene has a 70% chance of being relevant.
  3. Elegant approaches for model selection and model averaging.
     - E.g., optimize λ or optimize grouping of w elements.
  4. Easy to relax IID assumption.
     - E.g., hierarchical Bayesian models for data from different sources.
  5. Bayesian optimization: fastest rates for some non-convex problems.
  6. Allows models with unknown/infinite number of parameters.
     - E.g., number of clusters or number of states in hidden Markov model.

- Why isn't everyone using this?
  - Philosophical: Some people don't like "subjective" prior.
  - Computational: Typically leads to nasty integration problems.

Coin Flipping Example: MAP Approach

- MAP vs. Bayesian for a simple coin flipping scenario:
  1. Our likelihood is a Bernoulli,
  $$p(H|\theta) = \theta.$$
  2. Our prior assumes that we are in one of two scenarios:
     - The coin has a 50% chance of being fair ($\theta = 0.5$).
     - The coin has a 50% chance of being rigged ($\theta = 1$).
  3. Our data consists of three consecutive heads: 'HHH'.

- What is the probability that the next toss is a head?
  - MAP estimate is $\hat{\theta} = 1$, since $p(\theta = 1|HHH) > p(\theta = 0.5|HHH)$.
  - So MAP says the probability is 1.
  - But MAP overfits: we believed there was a 50% chance the coin is fair.

Coin Flipping Example: Posterior Distribution

- Bayesian method needs posterior probability over θ,

$$\begin{aligned} p(\theta = 1|HHH) &= \frac{p(HHH|\theta = 1)p(\theta = 1)}{p(HHH)} \\ &= \frac{p(HHH|\theta = 1)p(\theta = 1)}{p(HHH|\theta = 0.5)p(\theta = 0.5) + p(HHH|\theta = 1)p(\theta = 1)} \\ &= \frac{(1)(0.5)}{(1/8)(0.5) + (1)(0.5)} = \frac{8}{9}, \end{aligned}$$

and similarly we have $p(\theta = 0.5|HHH) = \frac{1}{9}$.

- So given the data, we should believe with probability $\frac{8}{9}$ that coin is rigged.
  - But there is still a $\frac{1}{9}$ probability that it is fair.

Coin Flipping Example: Posterior Predictive

- Posterior predictive gives probability of head given data and prior,

$$p(H|HHH) = p(H, \theta = 1|HHH) + p(H, \theta = 0.5|HHH)$$
$$= p(H|\theta = 1, HHH)p(\theta = 1|HHH) + p(H|\theta = 0.5, HHH)p(\theta = 0.5|HHH)$$
$$= (1)(8/9) + (0.5)(1/9) = 0.94. \qquad = 0.94$$

- So the correct probability given our assumptions/data is 0.94, and not 1.

- Notice that there was no optimization of the parameter θ:
  - In Bayesian stats we condition on data and integrate over unknowns.

Coin Flipping Example: Discussion

Comments on coin flipping example:

- Bayesian prediction uses that HHH could come from fair coin.
- As we see more heads, posterior converges to 1.
  - ML/MLE/Bayes usually agree as data size increases.
- If we ever see a tail, posterior of $\theta = 1$ becomes 0.

- If the prior is correct, then Bayesian estimate is optimal:
  - Bayesian decision theory gives optimal action incorporating costs.
- If the prior is incorrect, Bayesian estimate may be worse.
  - This is where people get uncomfortable about "subjective" priors.
- But ML/MAP are also based on "subjective" assumptions.

Bayesian Model Averaging

- In 340 we saw that model averaging can improve performance.
  - E.g., random forests average over random trees that overfit.
- But should all models get equal weight?
  - What if we find a random stump that fits the data perfectly?
    - Should this get the same weight as deep random trees that likely overfit?
  - In science, research may be fraudulent or not based on evidence.
    - E.g., should we vaccines cause autism or climate change denial models?
- In these cases, nave averaging may do worse.

Bayesian Model Averaging

- Suppose we have a set of m probabilistic classifiers w_j
  - Previously our ensemble method gave all models equal weights,

$$p(\hat{y}|\hat{x}) = \frac{1}{m}p(\hat{y}|\hat{x}, w_1) + \frac{1}{m}p(\hat{y}|\hat{x}, w_2) + \cdots + \frac{1}{m}p(\hat{y}|\hat{x}, w_m).$$

- Bayesian model averaging weights by posterior,

$$p(\hat{y}|\hat{x}) = p(w_1|X, y)p(\hat{y}|\hat{x}, w_1) + p(w_2|X, y)(\hat{y}|\hat{x}, w_2) + \cdots + p(w_m|X, y)p(\hat{y}|\hat{x}, w_m).$$

- So we should weight by probability that w_j is the correct model.
  - Equal weights assume all models are equally probable.

Bayesian Model Averaging

- Weights are posterior, so proportional to likelihood times prior:

$$p(w_j|X,y) \propto p(y|X,w_j)p(w_j).$$

- Likelihood gives more weight to models that predict y well.
- Prior should gives less weight to models that are likely to overfit.

- This is how rules of probability say we should weight models.
  - It's annoying that it requires a "prior" belief over models.
  - But as $n \to \infty$, all weight goes to "correct" model[s] w^* as long as $p(w^*) > 0$.

6 Ingredients of Bayesian Inference

1. Likelihood $p(y|X, w)$.
   - Probability of seeing data given parameters.
2. Prior $p(w|\lambda)$.
   - Belief that parameters are correct before we've seen data.
3. Posterior $p(w|X, y, \lambda)$.
   - Probability that parameters are correct after we've seen data.
   - We won't use the MAP "point estimate", we want the whole distribution.

6 Ingredients of Bayesian Inference

4. Posterior predictive $p(\hat{y}|\hat{x}, X, y, \lambda)$.
   - Probability of new data given old, integrating over parameters.
   - This tells us which prediction is most likely given data and prior.
5. Marginal likelihood $p(y|X, \lambda)$ (also called evidence).
   - Probability of seeing data given hyper-parameters.
   - We'll use this later for setting hyper-parameters.
6. Cost $C(\hat{y}|\tilde{y})$.
   - The penalty you pay for predicting $\hat{y}$ when it was really was $\tilde{y}$.
   - Leads to Bayesian decision theory: predict to minimize expected cost.

Summary

- Convolutional neural networks: unprecedented image classification performance.
  - Gradient descent on input images leads to inceptionism and artistic style transfer.
- Fully-convolutional networks:
  - Let us apply convolutional networks for structured prediction problems.
- Bayesian statistics:
  - Condition on data and integrate (rather than maximize) over posterior.

- Next time: we relax IID and get a different answer than "use cross-validation".