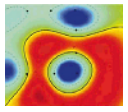
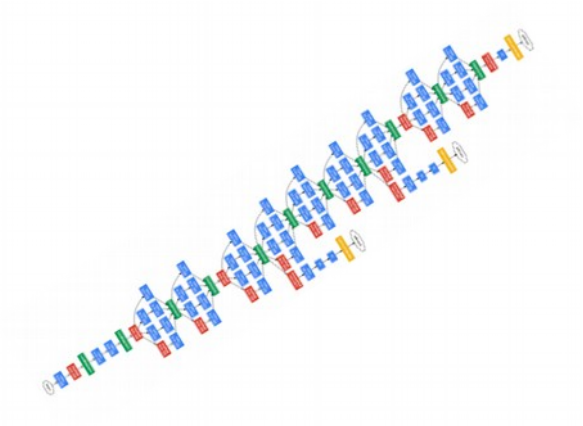
SoSe 2018: Deep Neural Networks

Lecture 4: Regularization

Machine Learning Group
Technische Universität Berlin



Deep Neural Networks: Wrap-Up

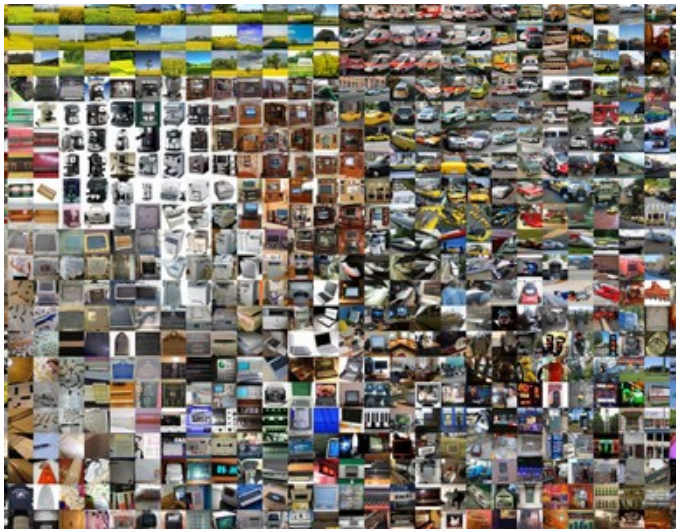


neural nets enable

big models

GPUs + fast algorithms

enable **training** these models



What about the training data?

IMAGENET

14,197,122 images, 21841 synsets indexed

Explore Download Challenges Publications CoolStuff About

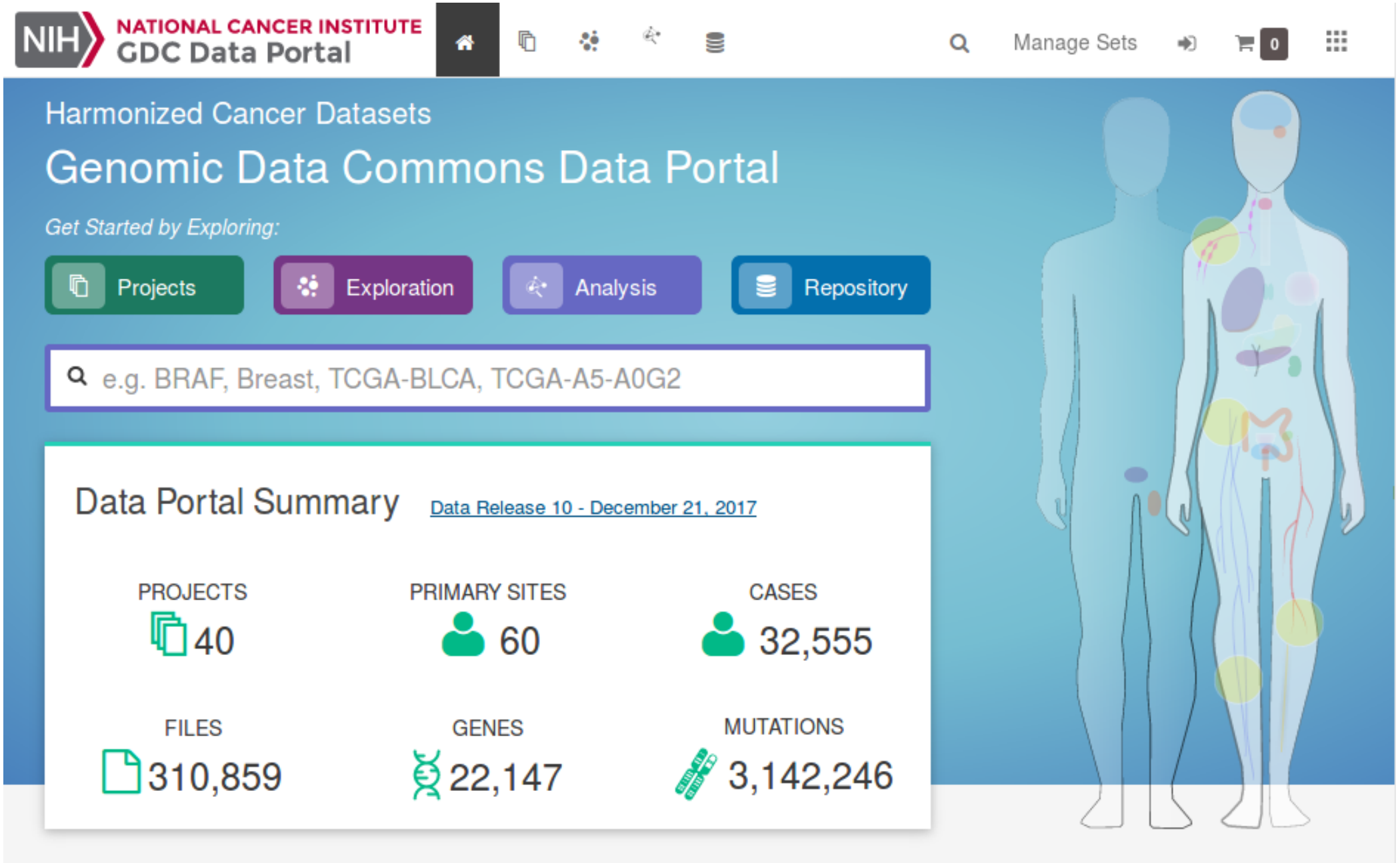
Not logged in. Login | Signup

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.

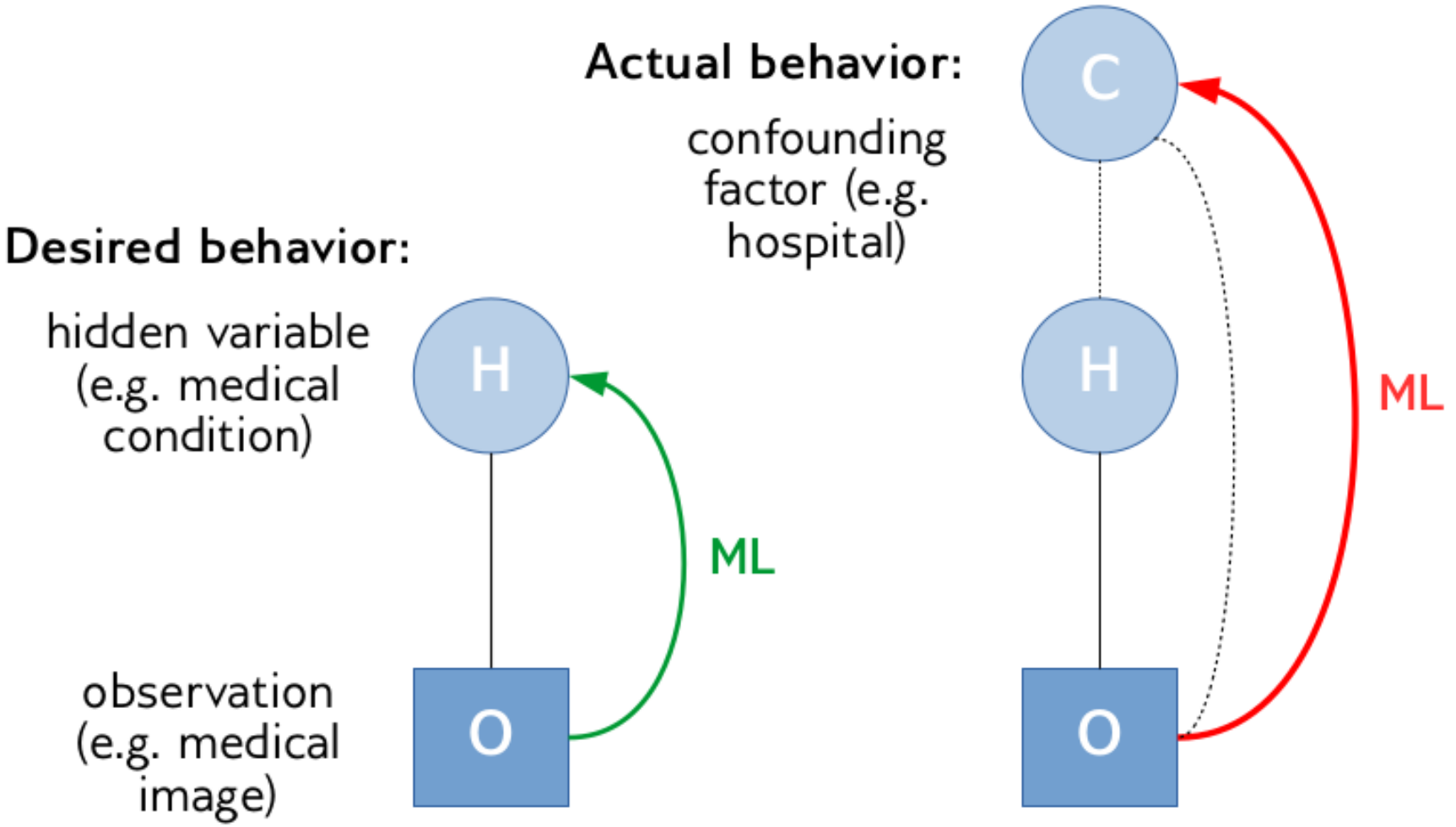
Click here to learn more about ImageNet, Click here to join the ImageNet mailing list.

What do these images have in common? *Find out!*

Check out the ImageNet Challenge on Kaggle!


NIH
NATIONAL CANCER INSTITUTE
GDC Data Portal
Manage Sets
0
Harmonized Cancer Datasets
Genomic Data Commons Data Portal
Get Started by Exploring:
Projects
Exploration
Analysis
Repository
e.g. BRAF, Breast, TCGA-BLCA, TCGA-A5-A0G2
Data Portal Summary
Data Release 10 - December 21, 2017
PROJECTS
40
PRIMARY SITES
60
CASES
32,555
FILES
310,859
GENES
22,147
MUTATIONS
3,142,246

A Common Problem: Confounding Factors



“Big Data” and “Big Confusion”

Observation: Many confounding factors in practice.

different qualities of measurement

different subjects

technology evolves over time

time of the day

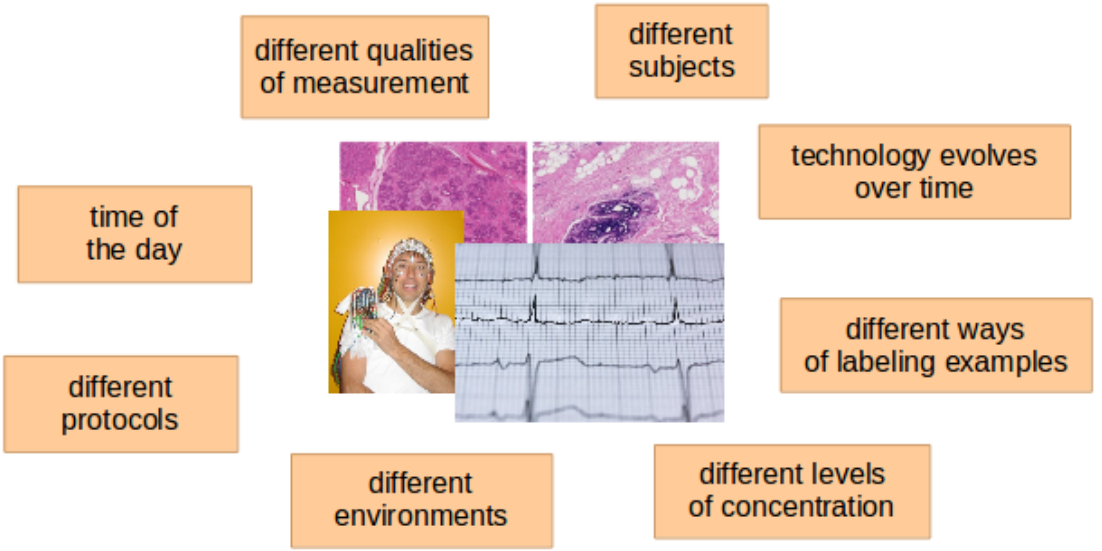
different ways of labeling examples

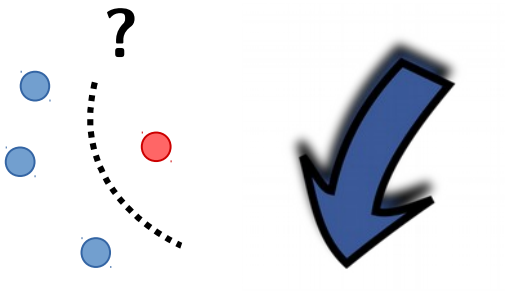
different protocols

different environments

different levels of concentration

Avoiding “Big Confusion”





Give up on big data, use a controlled setting, and make careful use of “little” data.

Inspect what the big data model has learned.

Occam's Razor

"Among competing hypotheses, the one with the fewest assumptions should be selected."

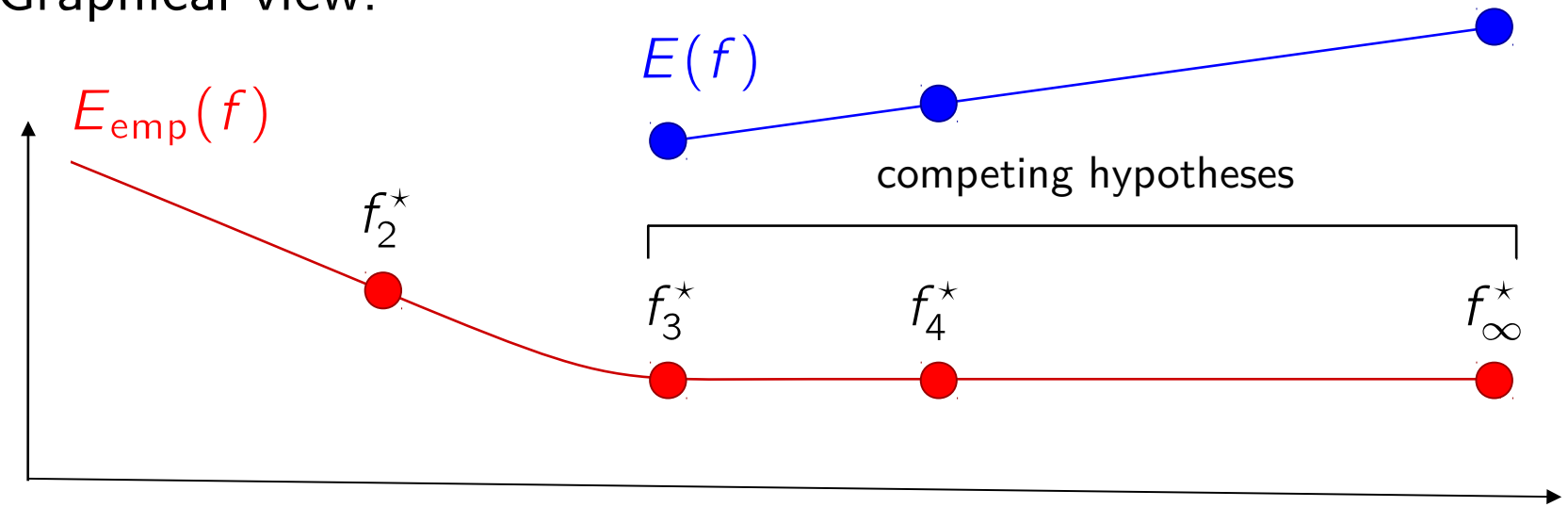
Domingos (1998): "Occam's two Razors: The Sharp and the Blunt":

Given two models with the same training-set error the simpler one should be preferred because it is likely to have lower generalization error."

Occam's Razor

"Among competing hypotheses, the one with the fewest assumptions should be selected."

Graphical view:



Complexity and Generalization Error

Generalization bound [Vapnik'95]:

Let h denote the VC-dimension of $\mathcal{F}$. The true error $E[f]$ (with $f \in \mathcal{F}$) is upper-bounded as:

$$E[f] \leq E_{\text{emp}}[f] + \sqrt{\frac{h\left(\log \frac{2N}{h} + 1\right) - \log(\eta/4)}{N}}$$

with probability $1 - \eta$.

Question:

- Can we compute the VC-dimension h?

VC-Dimension of Various Models

1. Model with a single parameter:

VC Dimension

$$f(x;\alpha) = \text{sign}(\sin(\alpha x)) \qquad h = \infty$$

complexity ≠ parameters

2. Linear models:

$$f(\boldsymbol{x}, (\boldsymbol{w}, b)) = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b) \qquad h = d + 1$$

Question: Can we make the data low-dimensional so that the VC dimension stays low?

Idea: Use unsupervised dimensionality reduction.

→ reduce dimensionality

PCA Dimensionality Reduction

Procedure:

1. Compute first *h* eigenvectors

$$\min_W \|X - WW^\top X\|^2 \qquad \text{s.t.} \quad W^T W = I$$

$$W \in \mathbb{R}^{d \times h} \qquad h \ll d$$

2. Project data on the first *h* eigenvectors

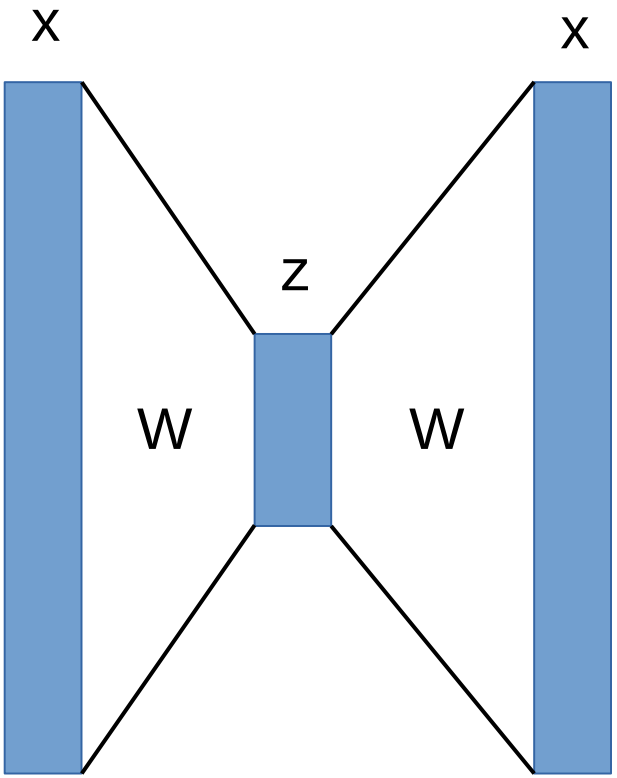
$$Z = W^\top X$$

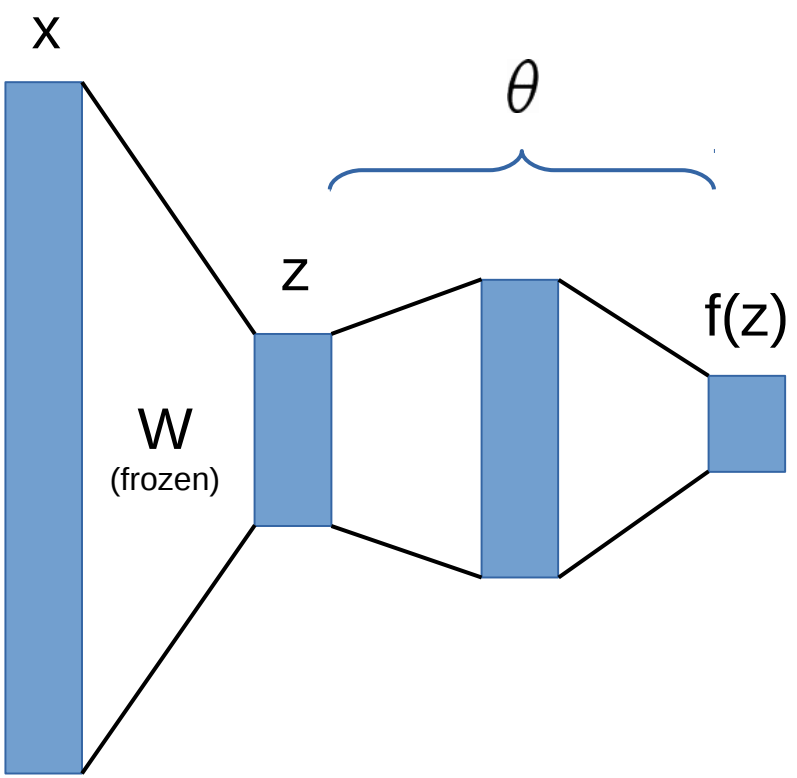
3. Train a neural network *f* on the low-dimensional data

$$\min_\theta \frac{1}{N} \sum_n \| f_\theta(z_n) - t_n \|^2$$

PCA Dimensionality Reduction

Graphical view:



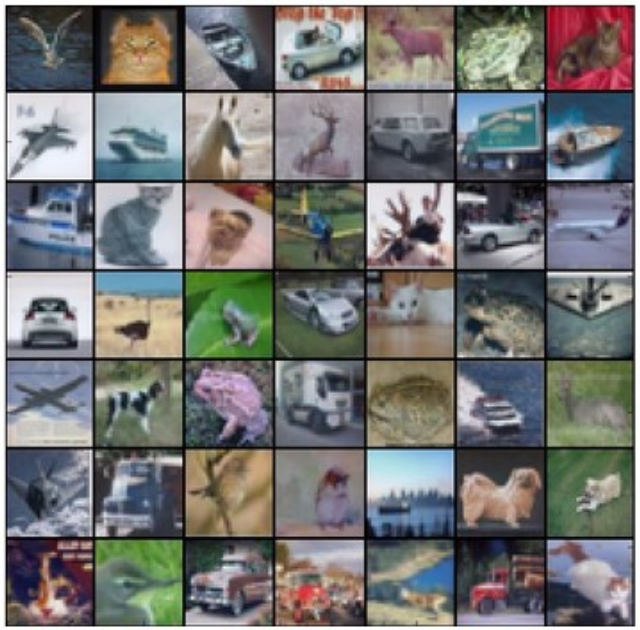


$$\min_W \|X - WW^\top X\|^2$$

$$\min_\theta \frac{1}{N} \sum_n \|f_\theta(z_n) - t_n\|^2$$

PCA Dimensionality Reduction

Example: Image classification



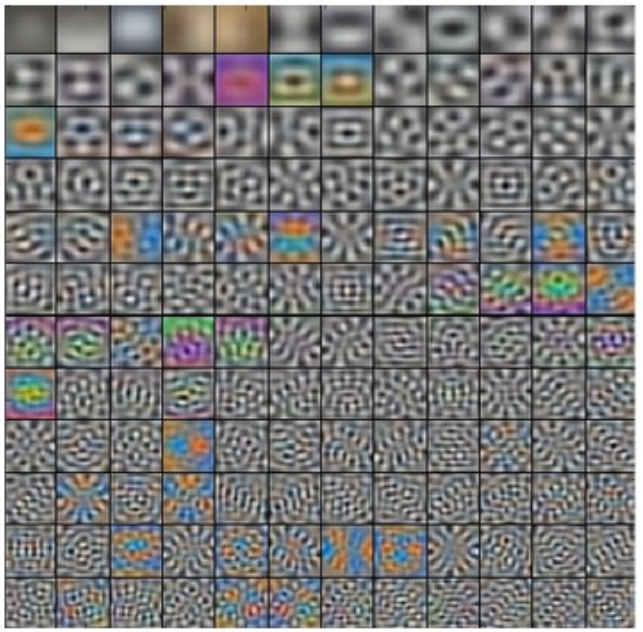


Image from Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition.

Observation: Projecting on the first principal components prevents the subsequent classifier to use high frequencies for classification.

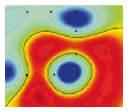
PCA Dimensionality Reduction

Advantages:

- Can build invariances into the classifier.
- If the neural network is constrained to be linear, the VC-dimension of the supervised model is reduced.

Disadvantages:

- PCA dimensionality reduction assumes that the discriminative information lies essentially in the leading PCA components (not true for images).
- Although the projected data is lower-dimensional, a neural network may still learn an uncontrollably complex classifier on this data (VC-dimension = ∞).



VC-Dimension of Various Models

1. Model with a single parameter:

$$f(x;\alpha) = \text{sign}(\sin(\alpha x))$$

VC Dimension

$$h = \infty$$

complexity ≠ parameters

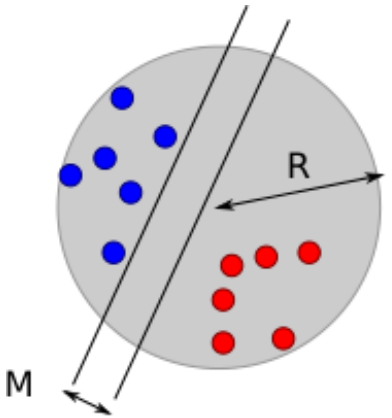
2. Linear models:

$$f(\boldsymbol{x}, (\boldsymbol{w}, b)) = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$$

$$h = d + 1$$

3. Linear models + large margin:

$$f(\boldsymbol{x}, (\boldsymbol{w}, b)) = \text{sign}(\boldsymbol{w}^\top \boldsymbol{x} + b)$$

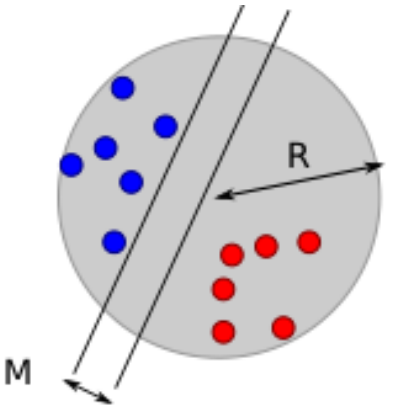


$$h = \min\left\{d + 1\,,\; 4\frac{R^2}{M^2} + 1\right\}$$

margin is key!

Large Margin → Reduce Gradient Norm

SVM Example: $f(\boldsymbol{x}) = \boldsymbol{x}^\top \boldsymbol{w} + b$



Objective

$$\min_{\boldsymbol{w},b} \|\boldsymbol{w}\|^2 \quad \text{s.t.} \quad \forall_{i=1}^{N} : y_i \cdot f(\boldsymbol{x}_i) \geq 1$$

Generalization to non-linear models (e.g. deep neural networks): minimize gradient (slope) of the discriminant function.

$$\min_{\theta} \left\| \frac{\partial f(\boldsymbol{x};\theta)}{\partial \boldsymbol{x}} \right\|^2$$

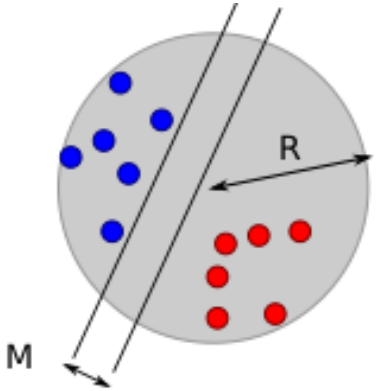
Questions:

- Gradient is different for different input values ***x***. Where should we enforce low gradient?

- Should large margin be enforced in input space or in representation space (like for kernels)?

Large Margin → Insensitivity to Perturbation

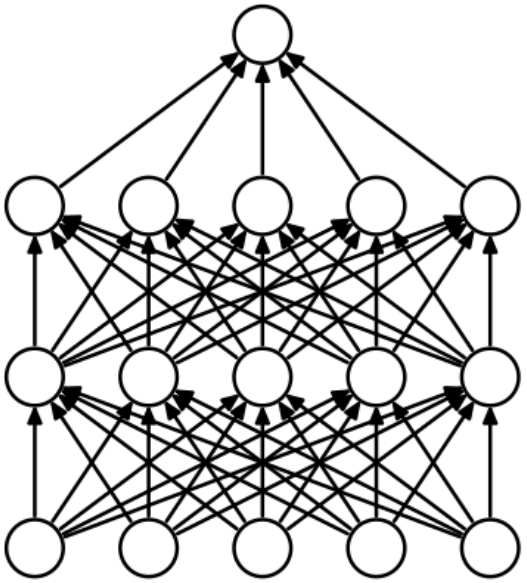
Linear Models: large-margin or ridge regularization makes the decision insensitive to small perturbation in the input space or feature space.



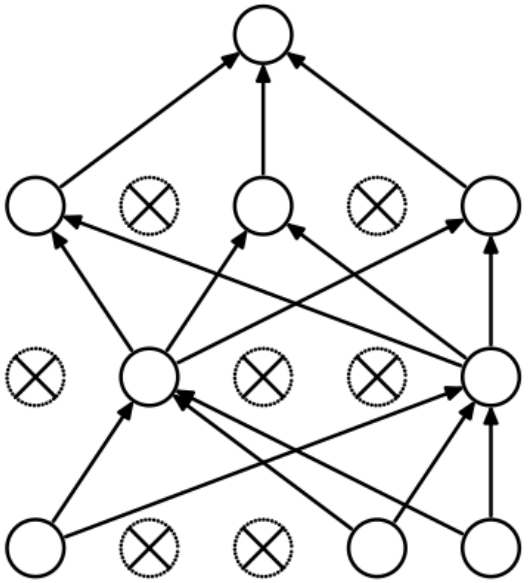
Neural networks: **Dropout algorithm** (Srivastava'14): replicates this insensitivity to perturbation in the context of neural network, by training the model while randomly turning on and off some neurons and input variables.

Example: The Dropout Method (Srivastava'14)

Neural networks: **Dropout algorithm** (Srivastava'14): replicates this insensitivity to perturbation in the context of neural network, by training the model while randomly turning on and off some neurons and input variables.



(a) Standard Neural Net

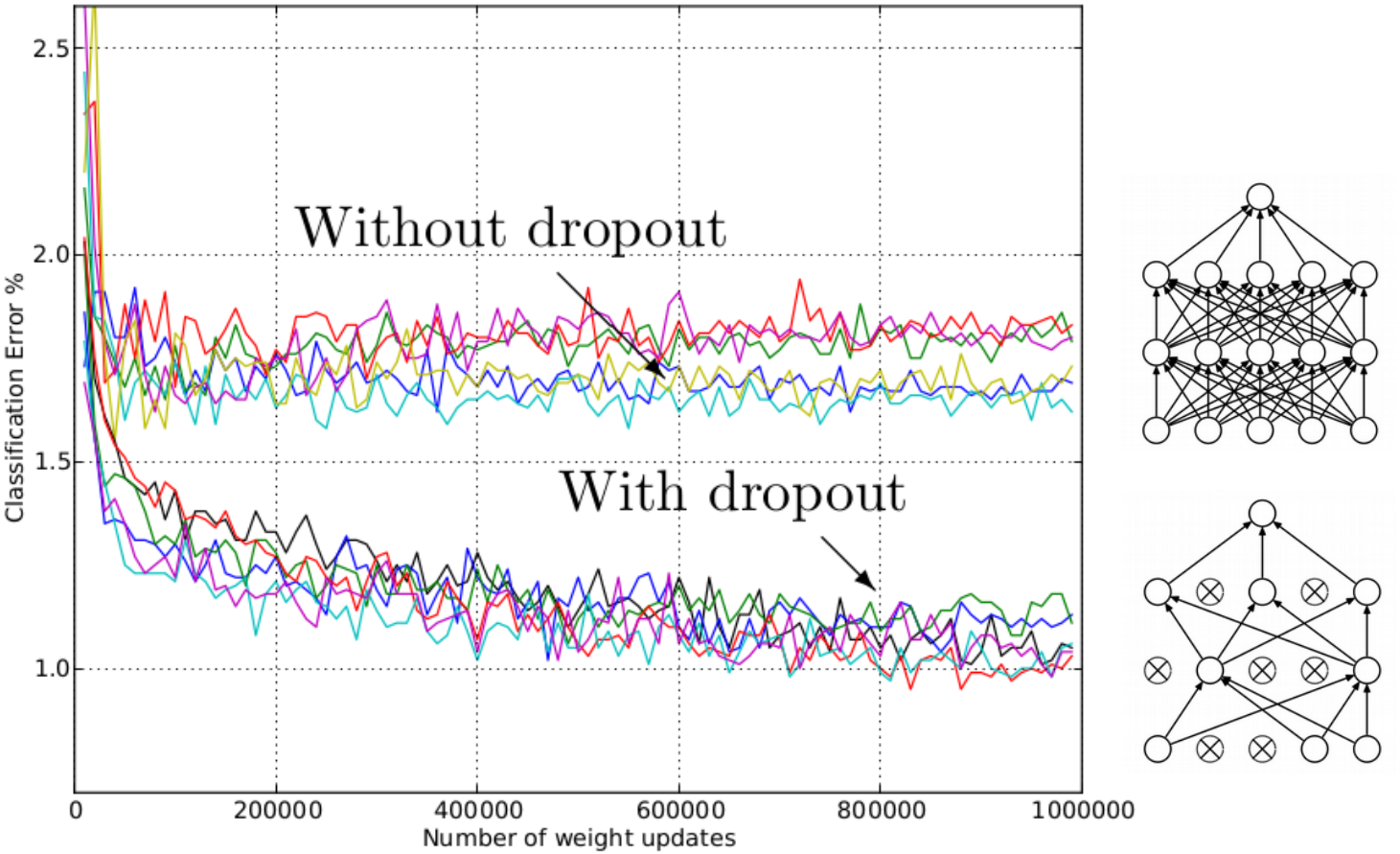


(b) After applying dropout.

(Source: Srivastava et al. 2014: Dropout: A simple way to prevent neural networks from overfitting).

Example: The Dropout Method (Srivastava'14)

Results on MNIST dataset:



(source: Srivastava et al. 2014: Dropout: A simple way to prevent neural networks from overfitting).

Other Approaches to Increase Margin

Learning with an adversary

$$\min_\theta \left[\frac{1}{N} \sum_n \| f_\theta(\boldsymbol{x}_n) - t_n \|^2 + \lambda \underbrace{\max_{\boldsymbol{x}} \frac{\| f_\theta(\boldsymbol{x}) - f_\theta(\boldsymbol{x}_n) \|^2}{\| \boldsymbol{x} - \boldsymbol{x}_n \|^2}}_{\text{adversary}} \right]$$

Advantages:
Look at the function more globally (not just based on the gradient or some specific perturbation).

Limitations:
- Min-max problems are harder to optimize.

Example of Min-Max approaches: Farnia et al.: A Minimax Approach to Supervised Learning. NIPS 2016.

Ensemble Learning

Idea: Train many neural networks, each of them starting from a different initialization and/or on a different subset of the training data. At test time, predict the test data with all networks and decide using e.g. majority voting.

Advantages:

- Can combine neural networks with different structures. Very easy to parallelize training.

Limitations:

- Need to train many models, and to ask each model for the prediction. → Resource intensive.

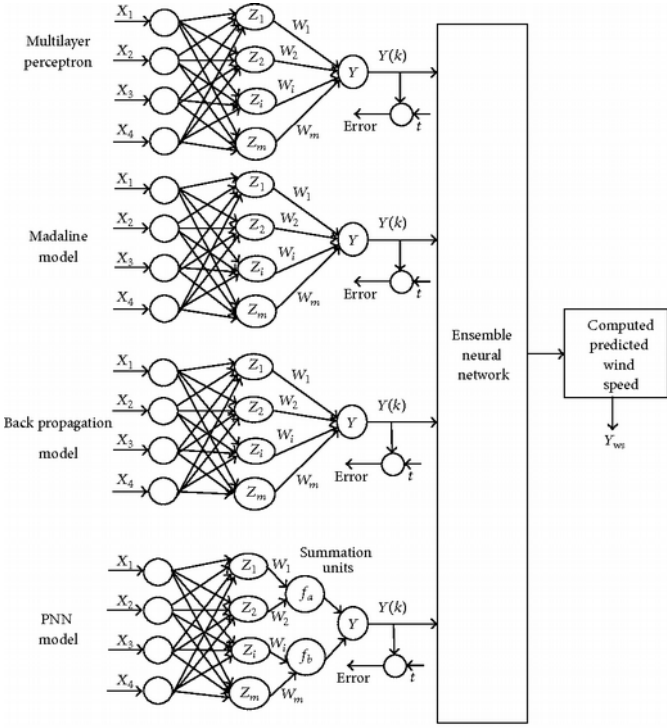


Image source: Ranganayaki et al. An Intelligent Ensemble Neural Network Model for Wind Speed Prediction in Renewable Energy Systems. 2016

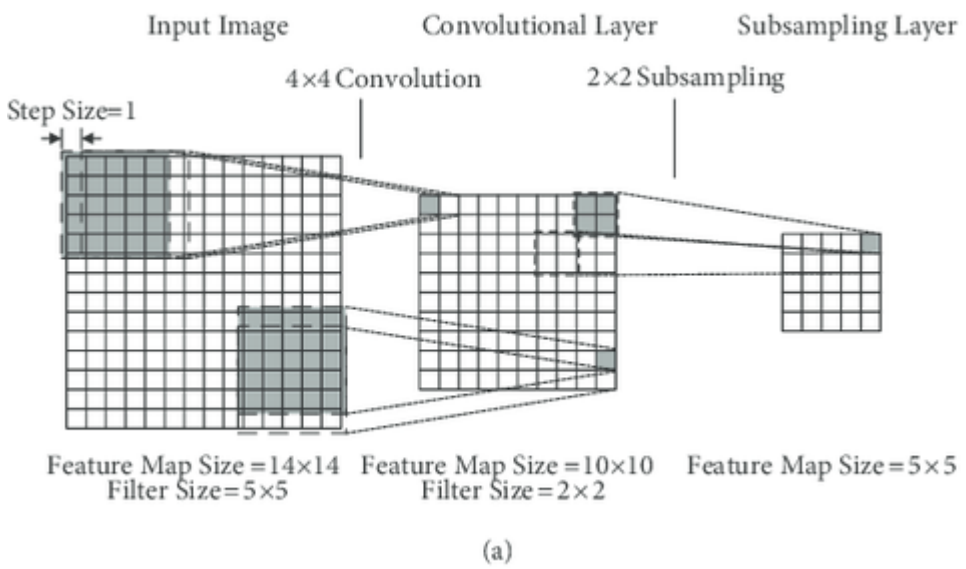
Insight: Dropout (c.f. previous slides) can be seen as an ensemble of $2^{\#\text{neurons}}$ networks.

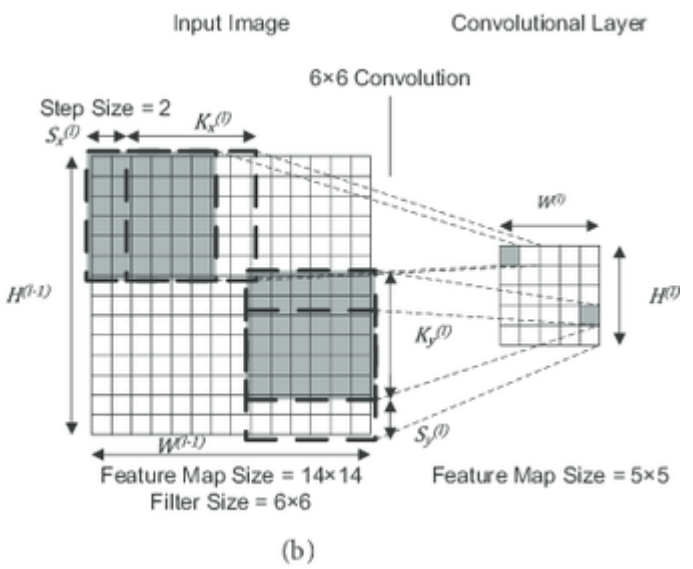
Question:

Can we use domain knowledge to further improve the supervised classifier?

Invariant Representations

Idea: Structure the network so that the prediction is structurally invariant to known class-preserving transformations. E.g. the convolutional neural network can implement translation invariance.



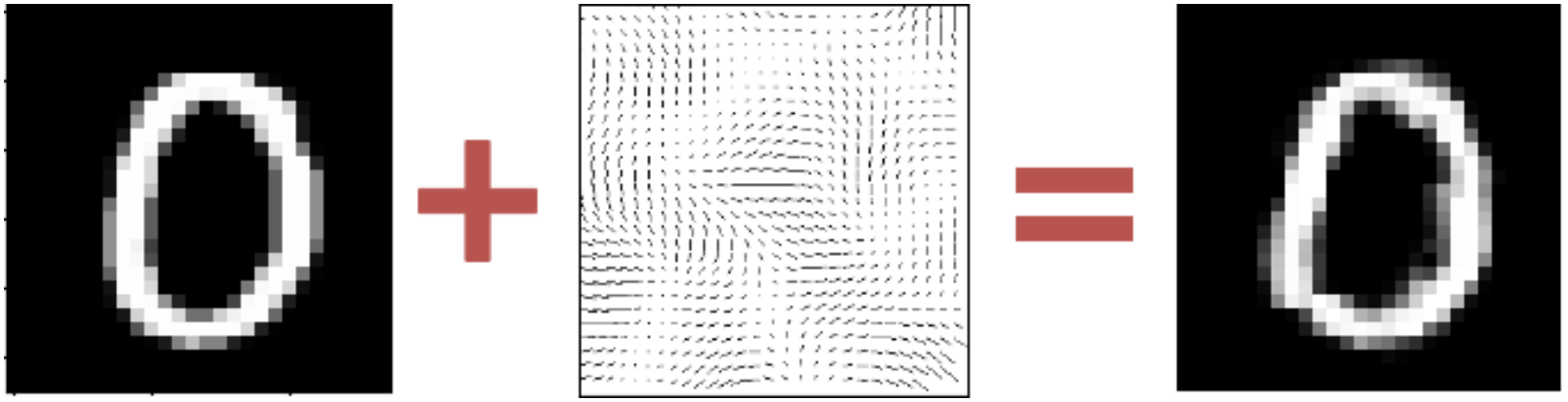


Source: Liew et al. 2016 Gender Classification: A Convolutional Neural Network Approach

Limitation: how to model complex invariances such as rotation?

Injecting Invariance

Invariance can be injected in the form of known class-preserving transformations in the input domain (e.g. elastic distortions for handwritten digits, rotations, or resizings/crops for images).

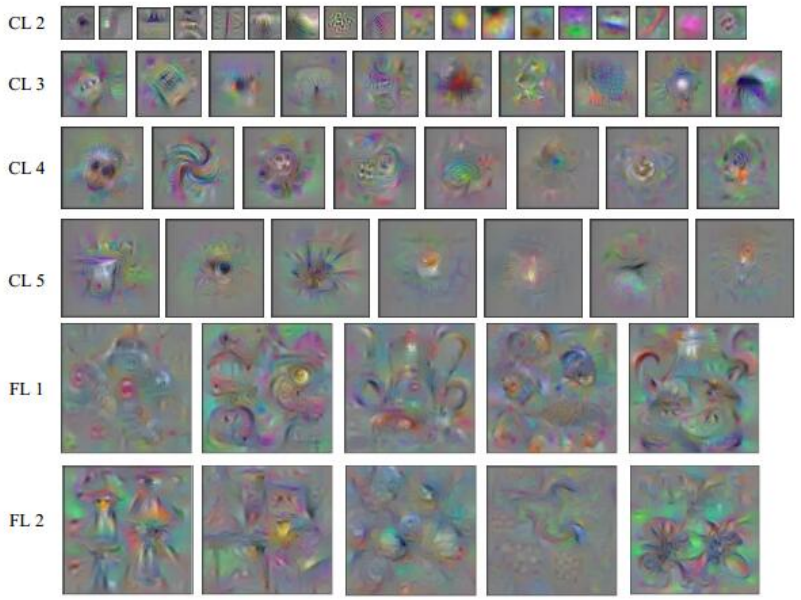


Source: M. Lerousseau “From Papers to Github #1: A practical guide to handwritten digits classifier & dataset preprocessing in Python and tensorflow”
https://marvinler.github.io/2017/03/11/from-papers-to-github-1.html

Transfer Learning

Transfer from a task where the network does not overfit (→can spend the time necessary to train complex representations). It is important that the transfer task is related to the target task.

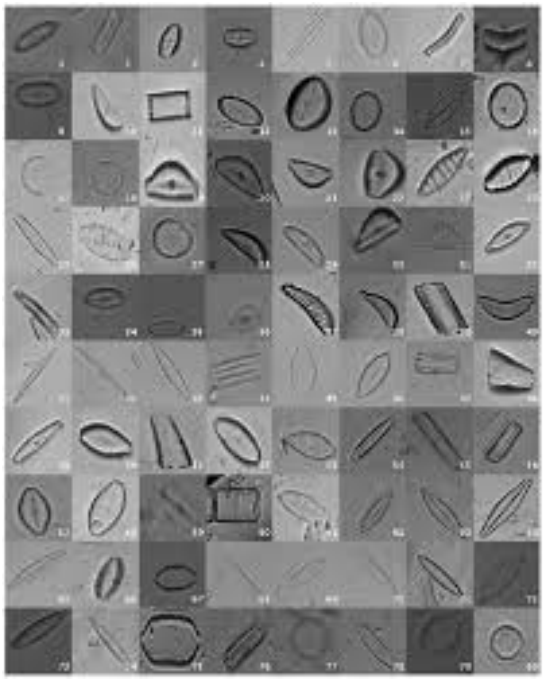
Lots of data and labels



neural network learn various concepts at different layers

Transfer parameters

Target task



few labels
(expensive to acquire)

Image Source: Pedraza'17: Automated Diatom Classification (Part B): A Deep Learning Approach

Image Source: Qin'18 How convolutional neural networks see the world (adapted from Yosinski'15)

Transfer Learning in Deep Networks

Once the network is trained, only the last few layers are fine-tuned on the target tasks.

Image Source: Mark Chang, Applied Deep Learning 11/03 Convolutional Neural Networks

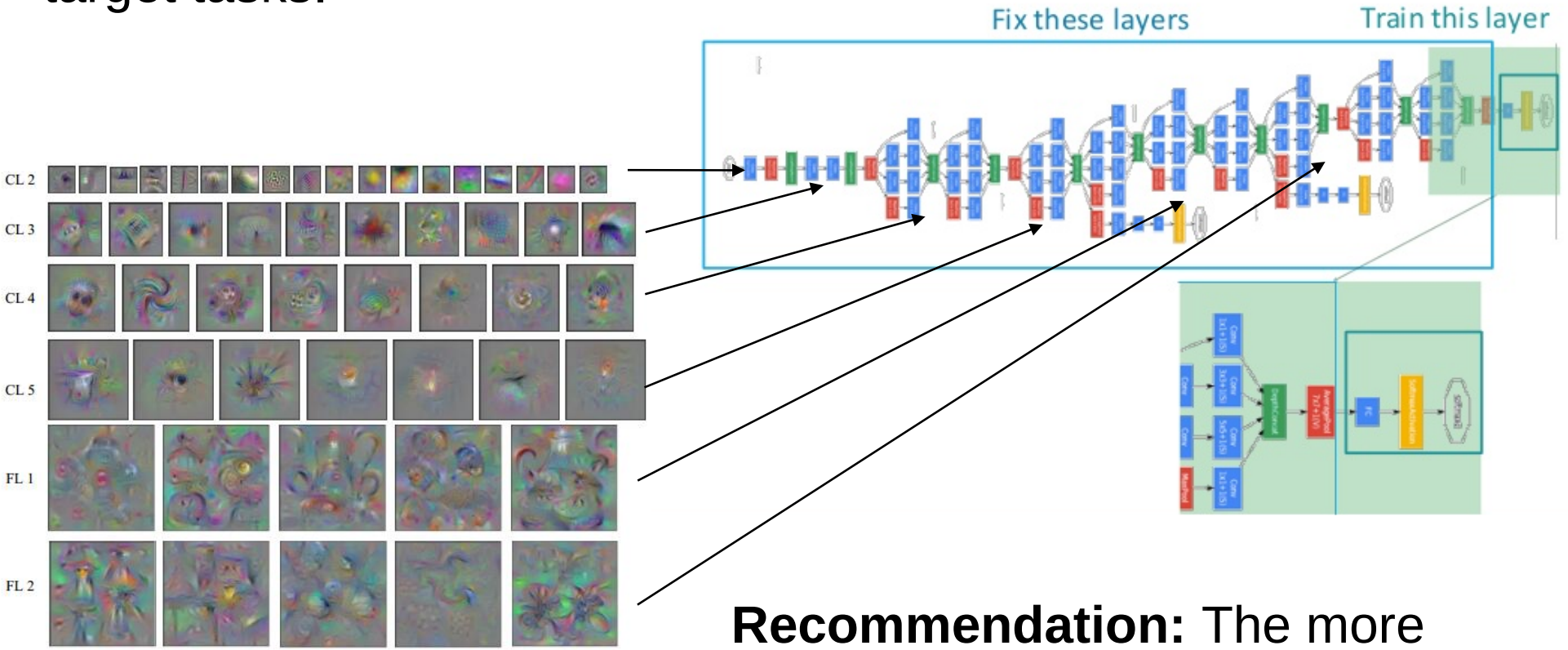


Image Source: Qin'18 How convolutional neural networks see the world (adapted from Yosinski'15)

Recommendation: The more similar the target task, the more layers we can keep.

Transfer Learning: Variants

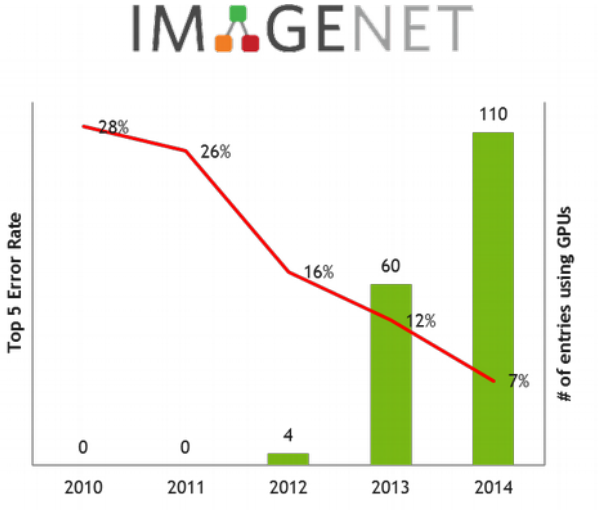
Many transfer learning tasks exist:

- **Unsupervised pretraining** (transfer from an unsupervised task such as learning to generate/reconstruct the data).
- **Many-tasks learning**. Training on many related task, all of which subject to overfitting, but when taken jointly, much less.
- Transfer from a **generic problem** with a lot of labeled data (e.g. general image classification, predicting Wikipedia text).
- **Meta-problem**: e.g. Learn to generate an associated image caption. Learn to predict next time frame. Learn to colorize the image. Learn to denoise, ...

The more related to the target task, the better.

Using Combination of Many Techniques

- Each technique (dropout, gradient regularization, ensembling, invariant architectures, injecting distortions, transfer learning, etc.) have their strengths and limitations.
- For maximum generalization, it is recommended to use several of these techniques at the same time. This is what people usually do when taking part to ML challenges (e.g. Kaggle challenges).



Summary

- Big data can sometimes have adverse effect on learned models (e.g. base its decision on spurious features).
- One approach to remediate this problem is to concentrate on a small dataset of “clean” data points, and make efficient use of this finite amount of data.
- Learning from finite examples is a well-studied machine learning problem with a lot of theory.
- While the theory is applicable to simpler models (e.g. linear), what the theory predicts (e.g. importance of margin) gives insight on how to improve deep networks.
- Building well-generalizing models also requires creativity (e.g. identifying and incorporating useful invariances, find related transfer tasks).