INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT-KERALA (IIITM-K)
An Autonomous Institute estd. by Govt. of Kerala
ഇന്ത്യൻ ഇൻസ്റ്റിറ്റ്യൂട്ട് ഓഫ് ഇൻഫർമേഷൻ ടെക്നോളജി ആൻറ് മാനേജ്മെൻറ്– കേരള, തിരുവനന്തപുരം

Machine Learning
and its applications

Oge Marques, PhD
Professor
College of Engineering and Computer Science
College of Business
Florida Atlantic University

FAU FLORIDA ATLANTIC UNIVERSITY

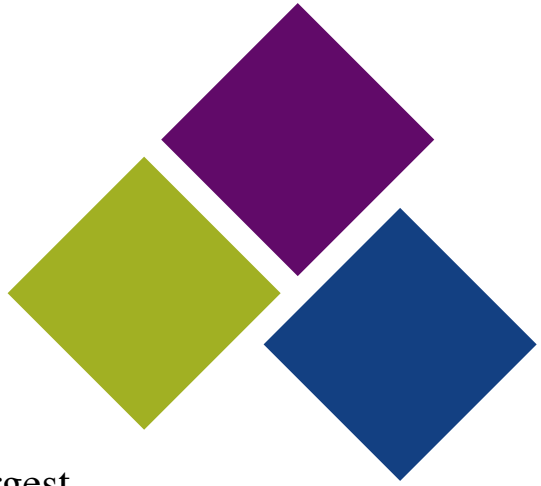
The Distinguished Speakers Program is made possible by

Association for
Computing Machinery

Advancing Computing as a Science & Profession

For additional information, please visit http://dsp.acm.org/

About ACM



ACM, the Association for Computing Machinery is the world's largest educational and scientific computing society, uniting educators, researchers and professionals to inspire dialogue, share resources and address the field's challenges.

ACM strengthens the computing profession's collective voice through strong leadership, promotion of the highest standards, and recognition of technical excellence.

ACM supports the professional growth of its members by providing opportunities for life-long learning, career development, and professional networking.

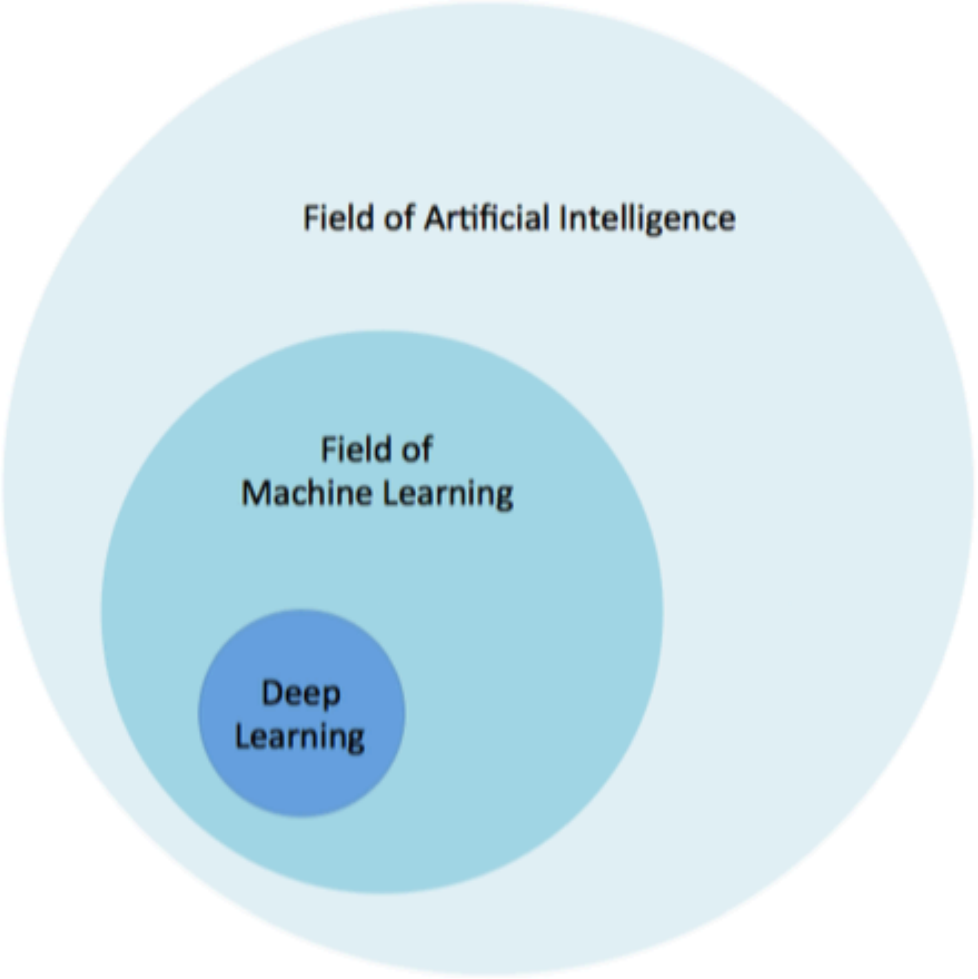
With over 100,000 members from over 100 countries, ACM works to advance computing as a science and a profession. www.acm.org

Part 1

Introduction

Introduction

- Machine learning has become the most visible and successful branch of artificial intelligence (AI).
- More recently, the emergence of a machine learning paradigm known as *deep learning* has enabled the development of intelligent systems that can, in many cases, demonstrate better-than-human performance.
- Many challenging research problems are being solved and new market opportunities have started to emerge.



Source: (Gibson&Patterson 2016)

The state of the art

- Self-driving cars
- DeepMind
- DeepFace
- AlphaGo
- OpenAI
- Deep Learning everywhere!

Part 2

Fundamentals of Machine Learning

Definition

What is Machine Learning?

- Machine learning teaches computers to do what comes naturally to humans and animals: learn from experience.
- Machine learning algorithms use computational methods to “learn” information directly from data without relying on a predetermined equation as a model.
- The algorithms adaptively improve their performance as the number of samples available for learning increases.

Source: (Mathworks 2016)

Applications

- Computational finance: credit scoring and algorithmic trading
- **Image processing and computer vision**: face recognition, motion detection, and object detection
- Computational biology: tumor detection, drug discovery, and DNA sequencing
- Energy production: price and load forecasting
- Automotive, aerospace, and manufacturing: predictive maintenance
- Natural language processing

Source: (Mathworks 2016)

Foundational concepts

- The meaning of ***learning***
  - An agent is learning if it improves its performance on future tasks after making observations about the world.
  - Any component of an agent can be improved by learning from data.
  - The improvements, and the techniques used to make them, depend on 4 major factors:
    - Which component is to be improved.
    - What prior knowledge the agent already has.
    - What representation is used for the data and the component.
    - **What feedback is available to learn from**.

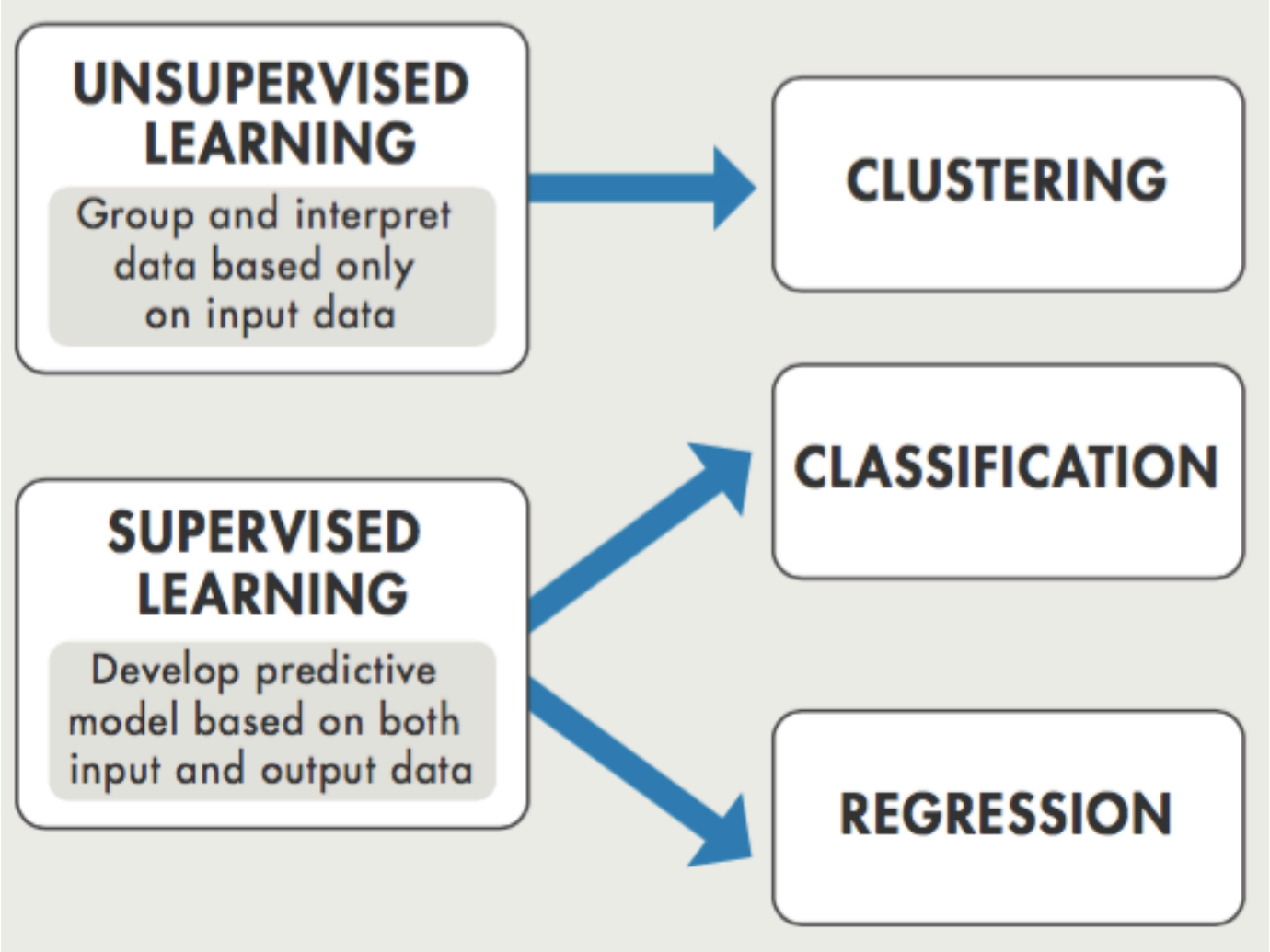
Source: (Russell & Norvig 2011)

Foundational concepts

- The types of feedback determine the main types of learning:
  - In **unsupervised learning** the agent learns patterns in the input even though no explicit feedback is supplied. Example: clustering.
  - In **reinforcement learning** the agent learns from a series of reinforcements—rewards or punishments.
  - In **supervised learning** the agent observes some example input–output pairs and learns a function that maps from input to output.
  - In **semi-supervised learning** we are given a few labeled examples and must make what we can of a large collection of unlabeled examples.

Source: (Russell & Norvig 2011)

Machine Learning techniques



Source: (Mathworks 2016)

Machine Learning techniques

- Supervised learning
  - **Classification techniques** predict *discrete* responses— for example, whether an email is genuine or spam, or whether a tumor is cancerous or benign.
    - Classification models classify input data into categories.
    - Typical applications: medical imaging, speech recognition, and credit scoring.
  - **Regression techniques** predict *continuous* responses— for example, changes in temperature or fluctuations in power demand.
    - Typical applications: electricity load forecasting and algorithmic trading.

Source: (Mathworks 2016)

Machine Learning techniques

- Unsupervised learning
  - Unsupervised learning finds hidden patterns or intrinsic structures in data.
    - It is used to draw inferences from datasets consisting of input data without labeled responses.
  - **Clustering** is the most common unsupervised learning technique.
    - It is used for exploratory data analysis to find hidden patterns or groupings in data.
    - Applications for clustering include gene sequence analysis, market research, and object recognition.

Source: (Mathworks 2016)

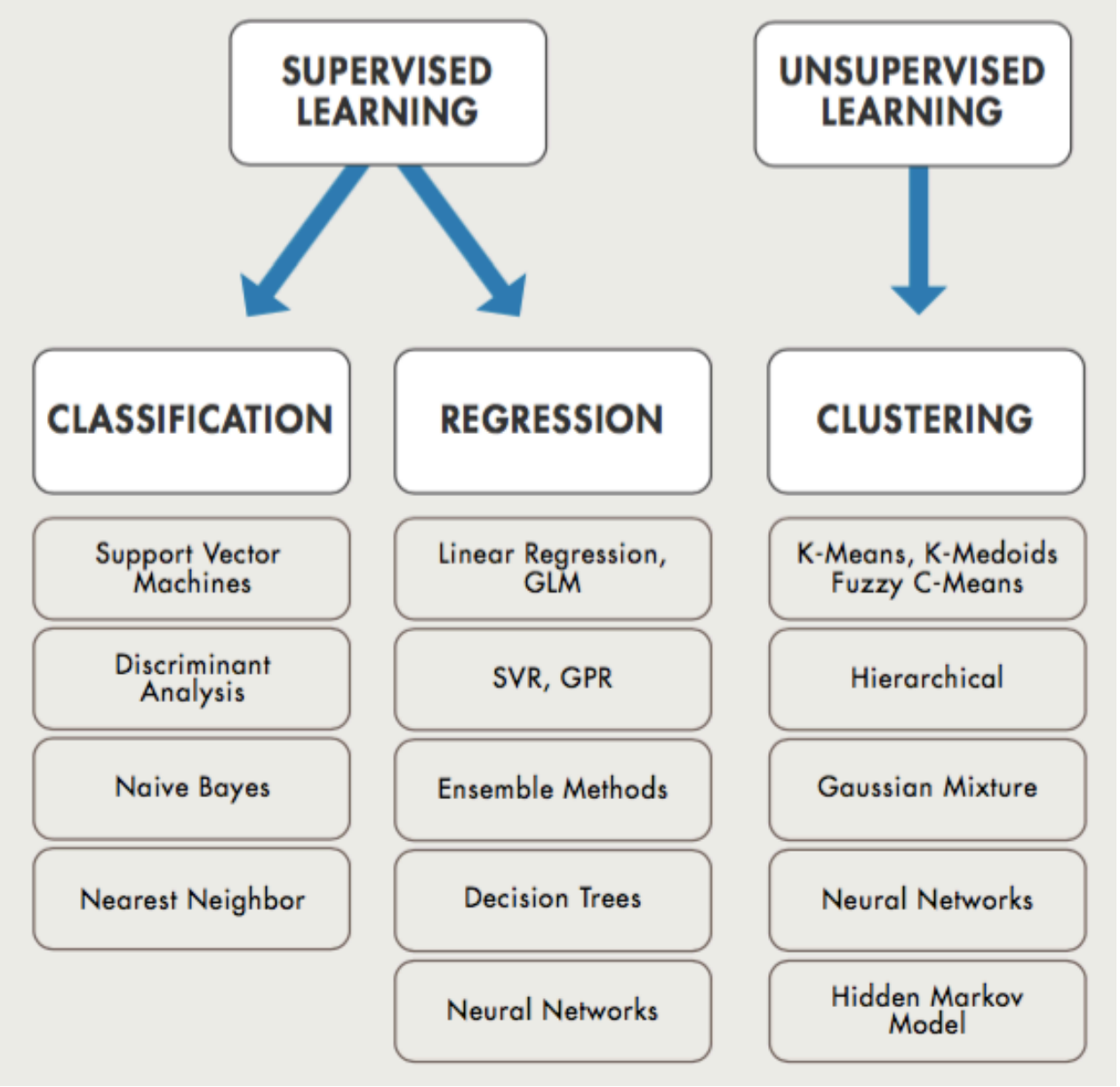
Machine Learning techniques

- Which algorithm to use?
  - A potentially overwhelming task!
    - There are dozens of supervised and unsupervised machine learning algorithms, and each takes a different approach to learning.
  - There is no best method or one size fits all.
    - Finding the right algorithm is partly just trial and error—even highly experienced data scientists can't tell whether an algorithm will work without trying it out.
  - But algorithm selection also depends on the size and type of data you're working with, the insights you want to get from the data, and how those insights will be used.

Source: (Mathworks 2016)

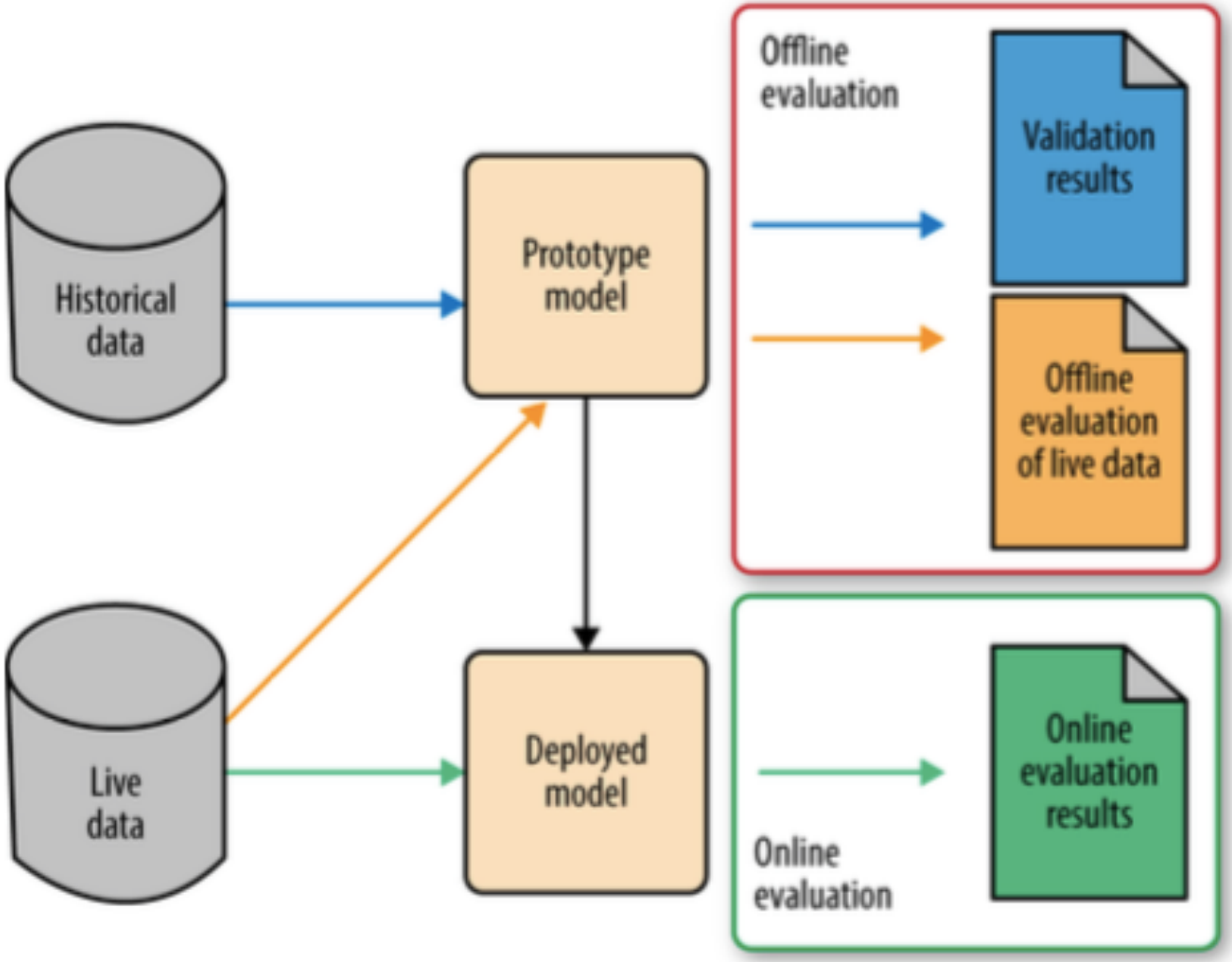
Machine Learning techniques

Which algorithm to use?



Source: (Mathworks 2016)

The ML workflow



Source: (Zheng 2015)

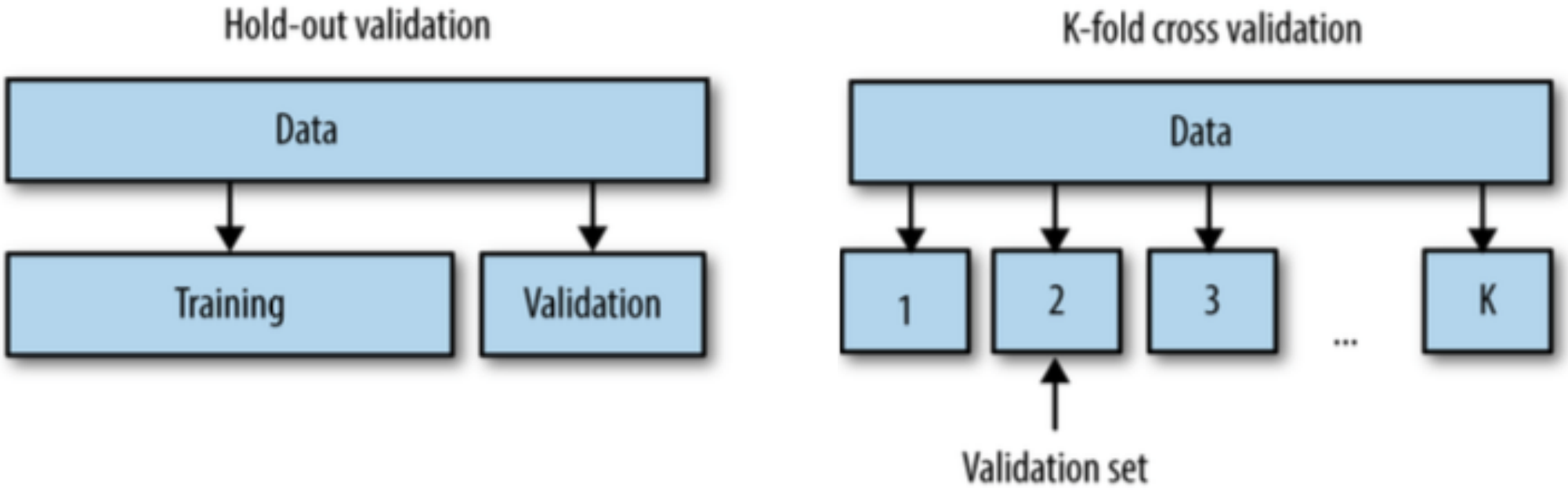
The ML workflow

- **Training set** — Which you run your learning algorithm on.
- **Dev (development) set** — Which you use to tune parameters, select features, and make other decisions regarding the learning algorithm. Sometimes also called the **(hold-out cross) validation set**.
- **Test set** — Which you use to evaluate the performance of the algorithm, but not to make any decisions about regarding what learning algorithm or parameters to use.

Source: (Ng 2016)

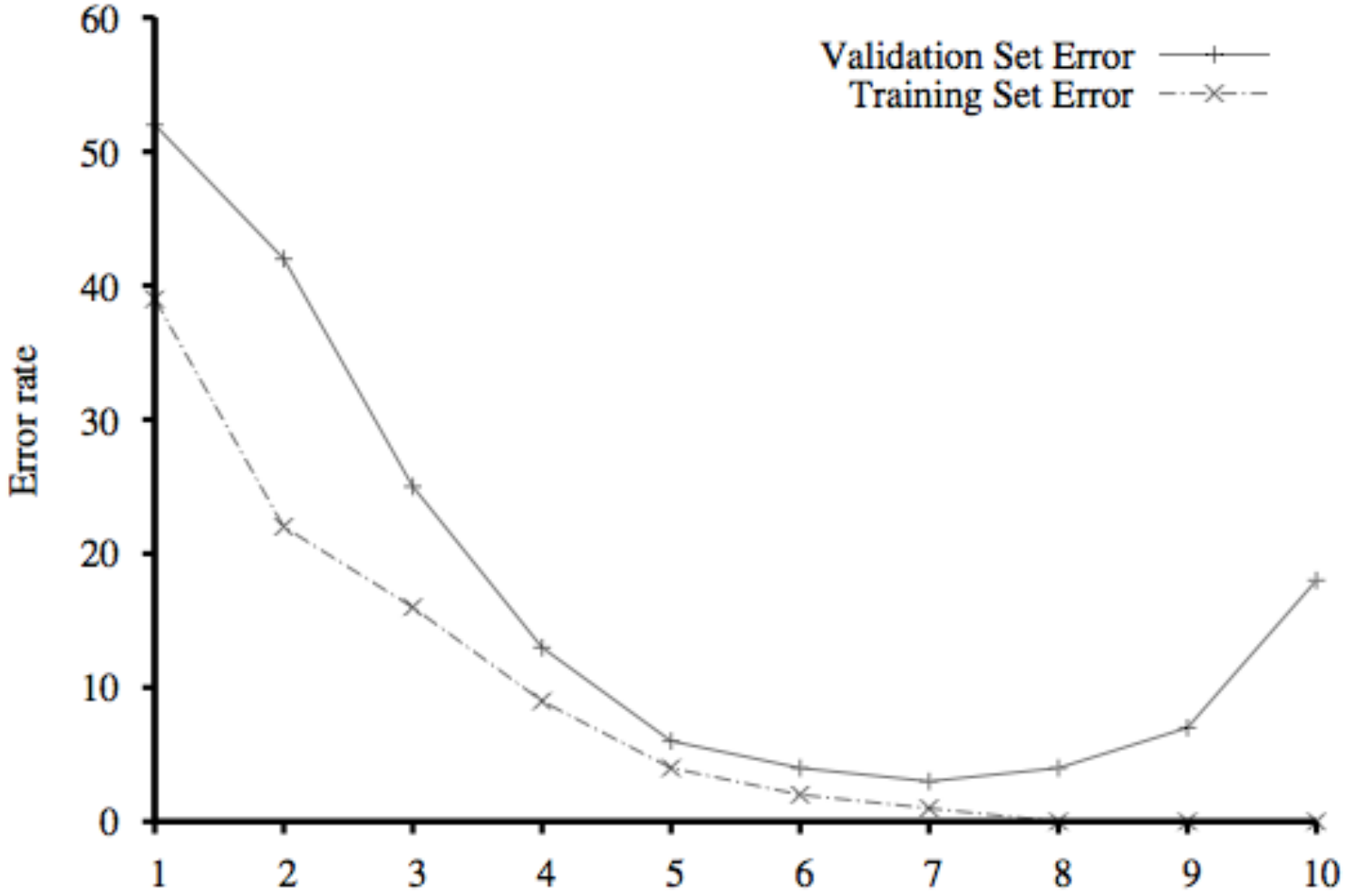
Validation methods

- Holdout validation vs. k-fold cross-validation



Source: (Zheng 2015)

Error rate in training and validation



Source: (Russell & Norvig 2011)

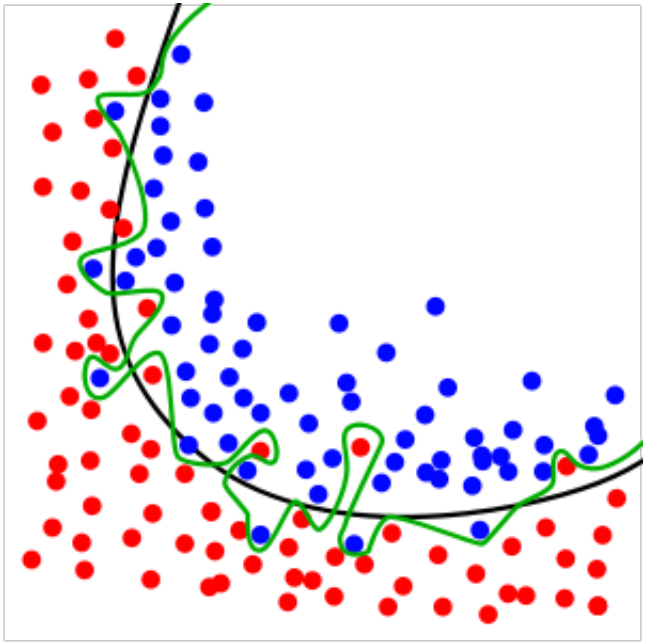
Loss function

- The loss function is defined as *the amount of utility lost by making an incorrect prediction*
  - Example (spam classifier)
    - If we believe that it is 10 times worse to classify non-spam as spam than vice-versa, then:
      - L(spam,nospam) = 1
      - L(nospam,spam) = 10
  - Common loss functions
    - absolute value of the difference (L_1 loss)
    - square of the difference (L_2 loss)

Source: (Russell & Norvig 2011)

Overfitting

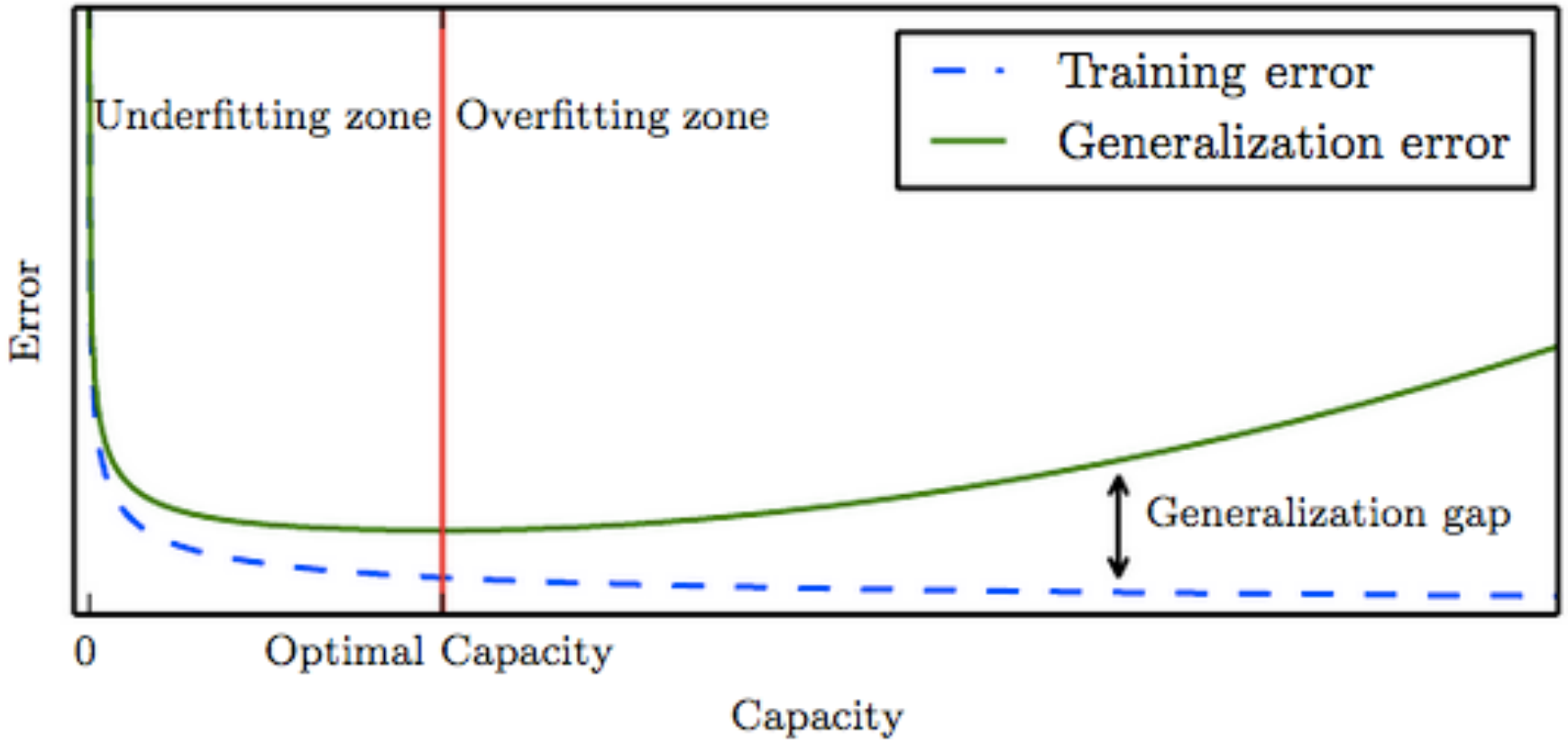
- If a model fits the training data too well, we have a problem called “overfitting”.
  - In this case, the model may have a low error rate for the training data but it does not generalize well to the overall population of data we’re interested in.



Source: (Gibson&Patterson 2016 | Wikimedia)

Overfitting

- Capacity and generalization gap



Source: (Goodfellow et al. 2016)

Evaluation criteria

- Confusion matrix

	P' (Predicted)	N' (Predicted)
P (Actual)	True Positive	False Negative
N (Actual)	False Positive	True Negative

We measure these answers by counting the number of:

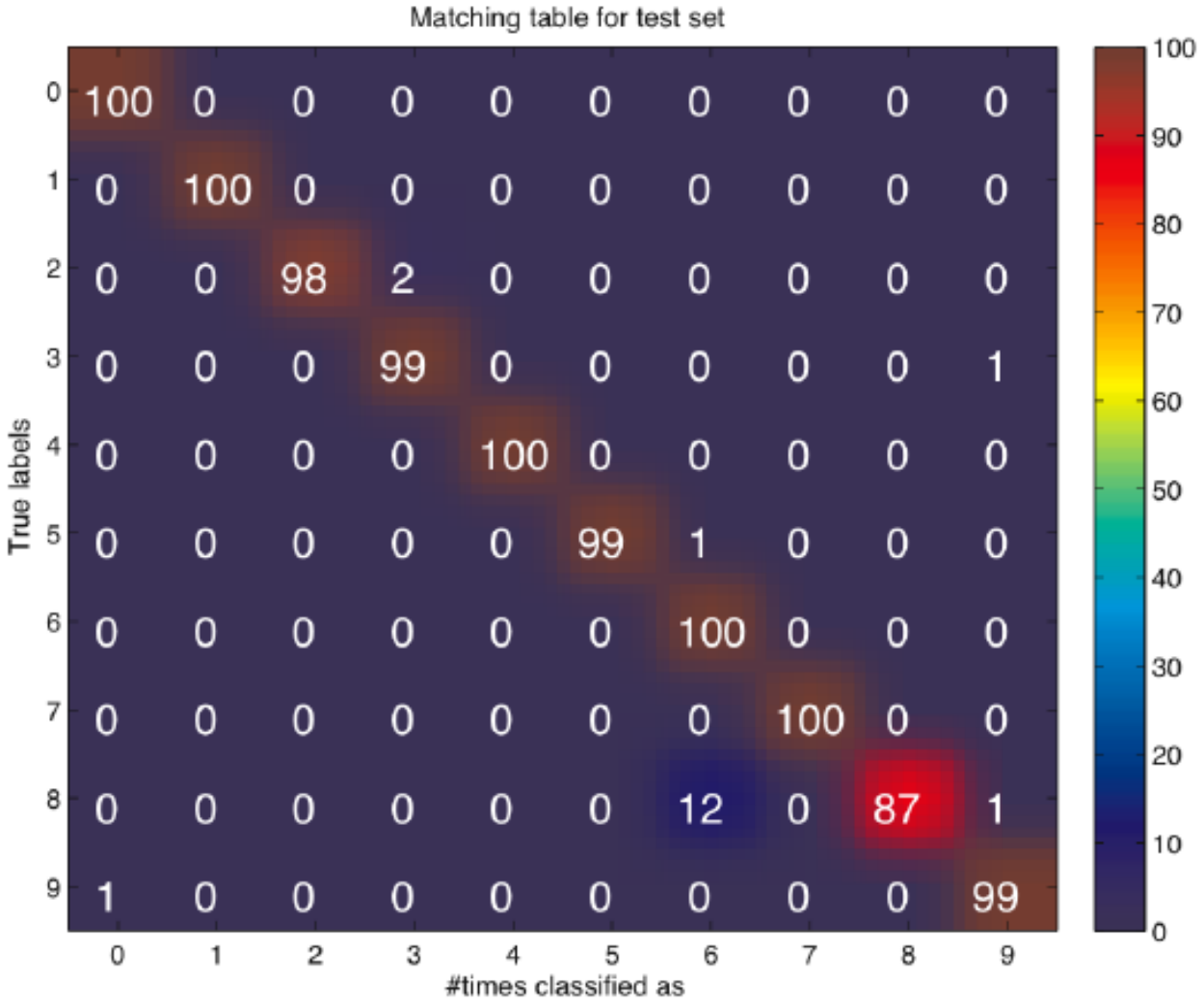
true positives
— positive prediction
— label was positive
false positives
— positive prediction
— label was negative
true negatives
— negative prediction
— label was negative
false negatives
— negative prediction
— label was positive

Source: (Gibson&Patterson 2016)

Evaluation criteria

- Confusion matrix: example

10-digit classifier (OCR)



Source: (Marques 2011)

Evaluation criteria

- Sensitivity, specificity, and accuracy
  - **Sensitivity** quantifies how well the model avoids false negatives.

$$\text{Sensitivity} = TP / (TP + FN)$$

  - **Specificity** quantifies how well the model avoids false positives.

$$\text{Specificity} = TN / (TN + FP)$$

  - **Accuracy** is the degree of closeness of measurements of a quantity to that quantity's true value.

$$\text{Accuracy} = (TP + TN) / (TP + FP + FN + TN)$$

Source: (Gibson&Patterson 2016)

Evaluation criteria

- Precision, recall, and F1
  - **Precision** (also known as *the positive prediction value*) is the degree to which repeated measurements under the same conditions give us the same results.

$$\text{Precision} = TP / (TP + FP)$$

  - **Recall** is the same as **sensitivity**

$$\text{Recall} = TP / (TP + FN)$$

  - The **F1 score** is the harmonic mean of both the precision and recall measures into a single score

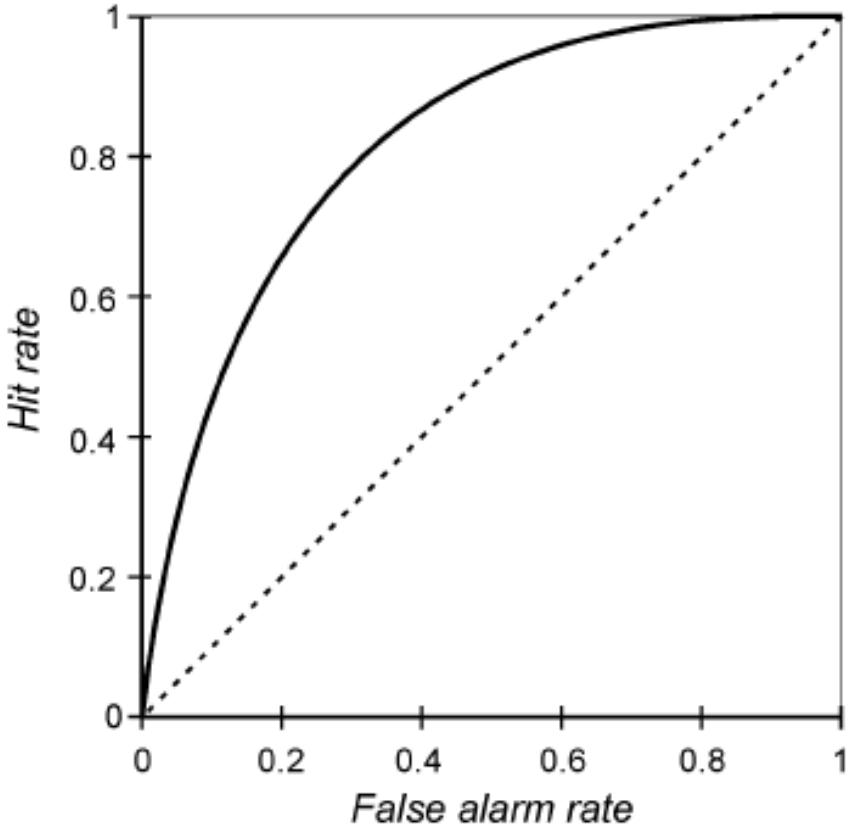
$$F1 = 2TP / (2TP + FP + FN)$$

Source: (Gibson&Patterson 2016)

Evaluation criteria

ROC (Receiver Operating Charateristic) curve and AUC (Area Under the Curve)

- The ROC curve is a plot that shows the relationship between the correct detection (true positive) rate (also known as hit rate) and the false alarm (false positive) rate.
- The ideal ROC curve is one in which the "knee" of the curve is as close to the top-left corner of the graph as possible, suggesting hit rate close to 100% with a false alarm rate close to zero.
- The AUC is a measure that allows for easy comparison among different ROCs (ideally, AUC = 1)



Source: (Marques 2011)

Practical hints and best practices

- Q: When should you use Machine Learning?
- A: When you have a *complex task or problem* involving a *large amount of data* and *lots of variables*, but *no existing formula or equation*.

- Machine learning is a good option if you need to handle situations like these:
  - *Hand-written rules and equations are too complex* — as in face recognition and speech recognition.
  - *The rules of a task are constantly changing* — as in fraud detection from transaction records.
  - *The nature of the data keeps changing, and the program needs to adapt* — as in automated trading, energy demand forecasting, predicting shopping trends

Source: (Mathworks 2016)

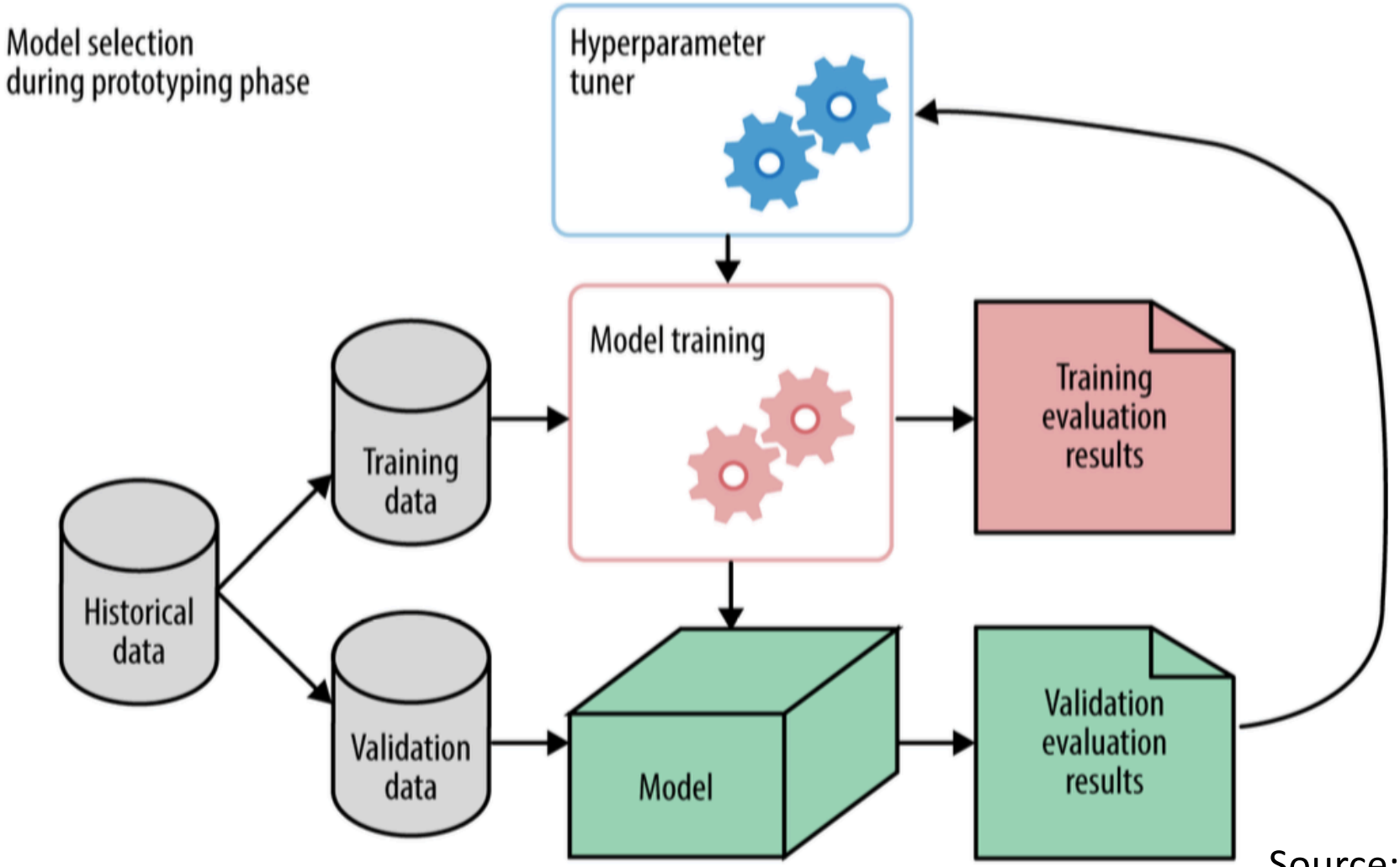
Practical hints and best practices

- Understanding the basics in applying machine learning is best framed by asking the correct questions to start with.
- We need to define:
  1. What is the **input data** we want to extract information (model) from?
  2. What kind of **model** is most appropriate for this data?
  3. What kind of **answer** would we like **to elicit from new data based on this model**?
- If we can answer these 3 questions we can setup a machine learning workflow that will build our model and produce our desired answers.

Source: (Gibson&Patterson 2016)

ML workflow

- The prototyping phase of building a ML model



Source: (Zheng 2015)

Practical hints and best practices

- Setting up development and test sets
  - Choose dev and test sets from a distribution that reflects what data you expect to get in the future and want to do well on.
    - This may not be the same as your training data's distribution.
  - Choose dev and test sets from the same distribution, if possible.
  - The old heuristic of a 70%/30% train/test split does not apply for problems where you have a lot of data; the dev and test sets can be much less than 30% of the data.
  - Your dev set should be large enough to detect meaningful changes in the accuracy of your algorithm, but not necessarily much larger.
  - Your test set should be big enough to give you a confident estimate of the final performance of your system.

Source: (Ng 2016)

Practical hints and best practices

- Beware of peeking!
  - Do not use the test set to make any decisions regarding the algorithm, including whether to roll back to the previous week's system.
    - If you do so, you will start to **overfit** to the test set, and can no longer count on it to give a completely unbiased estimate of your system's performance.

Source: (Ng 2016)

Practical hints and best practices

- Consider having a **single-number evaluation metric** (such as accuracy)
  - It allows you to sort all your models according to their performance on this metric, and quickly decide what is working best.
  - It speeds up your ability to make a decision when you are selecting among a large number of classifiers.
  - It gives a clear preference ranking among all of them, and therefore a clear direction for progress.

Source: (Ng 2016)

Practical hints and best practices

- Q: Which model is best?
- A: Classifier A

Classifier	Precision	Recall
A	95%	90%
B	98%	85%

Classifier	Precision	Recall	F1 score
A	95%	90%	**92.4%**
B	98%	85%	**91.0%**

Source: (Ng 2016)

Practical hints and best practices

- Q: Which model is best?
- A: It depends...
  - If running time < 100 ms = "satisficing metric"
  - Then Classifier B is best according to the "optimizing metric" (accuracy)

Classifier	Accuracy	Running time
A	90%	80ms
B	92%	95ms
C	95%	1,500ms

Source: (Ng 2016)

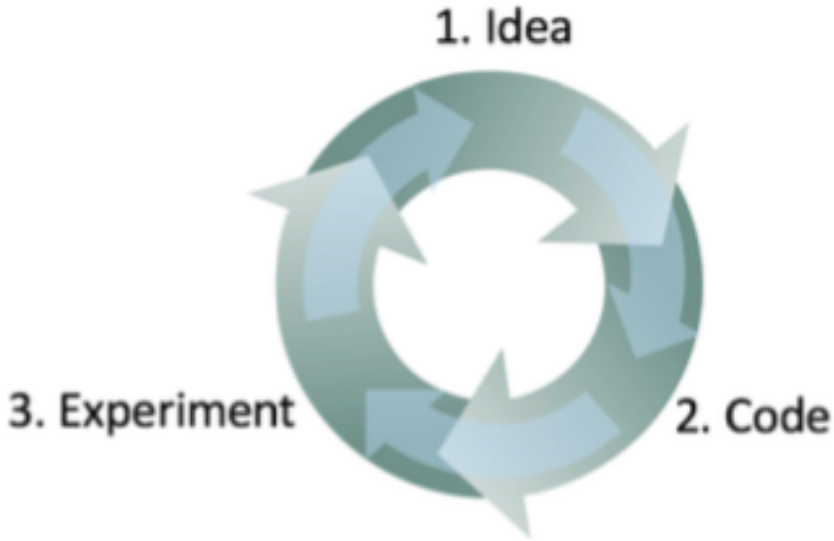
Practical hints and best practices

- If you are trading off N different criteria:
  - set N-1 of the criteria as “satisficing” metrics, i.e., you simply require that they meet a certain value.
  - then define the final one as the “optimizing” metric.

Source: (Ng 2016)

Practical hints and best practices

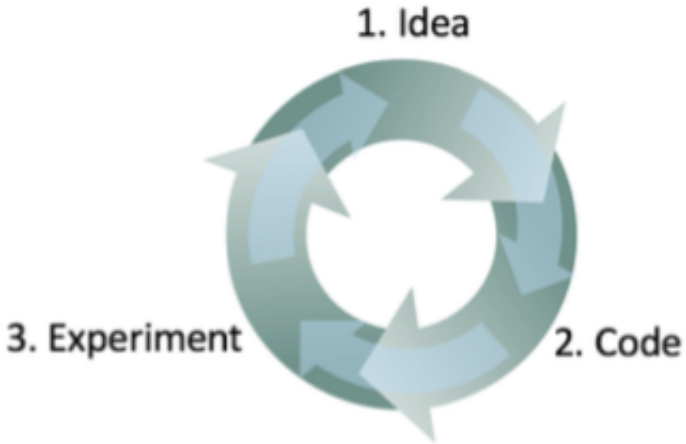
- Building a machine learning system is an iterative process:
  1. Start off with some **idea** on how to build the system.
  2. Implement the idea in **code**.
  3. Carry out an **experiment** which tells how well the idea worked.
     - Based on these learnings, go back to generate more ideas, and keep on iterating.



Source: (Ng 2016)

Practical hints and best practices

- Having a dev set and metric speeds up iterations
  - The faster you can go round this loop, the faster you will make progress.
  - This is why having dev/test sets and a metric are important:
    - Each time you try an idea, measuring your idea's performance on the dev set lets you quickly decide if you're heading in the right direction.



Source: (Ng 2016)

Practical hints and best practices

- If ever your dev set and metric are no longer pointing your team in the right direction, quickly change them:
  - (i) If you had overfit the dev set, get more dev set data.
  - (ii) If the actual distribution you care about is different from the dev/test set distribution, get new dev/test set data.
  - (iii) If your metric is no longer measuring what is most important to you, change the metric.

Source: (Ng 2016)

Practical hints and best practices

- Invest time into Error Analysis
  - "Error Analysis" refers to the process of (manually) examining dev set examples that your algorithm misclassified, so as to understand the underlying causes of the errors.
  - This can both help you prioritize projects and inspire new directions.
  - However, it does not result in a rigid mathematical formula that tells you what should be the highest priority task.

Source: (Ng 2016)

Practical hints and best practices

- Error Analysis example
  - Your cat detector solution has problems:
    1. Occasionally *dogs* are being recognized as cats.
    2. Sometimes "*great cats*" (lions, panthers, etc.) are recognized as house cats (pets).
    3. The system's performance on *blurry* images should be improved.
  - Which one would you tackle first?

Image	Dog	Great cat	Blurry	Comments
1	✔			Usual pitbull color
2			✔	
3		✔	✔	Lion; picture taken at zoo on rainy day
4		✔		Panther behind tree
...	...	...	...	...
% of total	8%	43%	61%	

Source: (Ng 2016)

A recipe for practitioners

1. Learn a (set of) tool(s) and framework(s)

   Candidates:

   – Python, NumPy, MatPlotLib, Pandas

   – R

   – MATLAB and its toolboxes (incl. MatConvNet)

   – WEKA and Java

   – Theano, TensorFlow, Keras (Deep Learning)

2. Get data

Source: (Brownlee 2016)

A recipe for practitioners

3. Understand the data

- Statistics
  - Distribution
  - Max, Min, Mean, StdDev, outliers
- Visualization
  - Histograms (of each attribute)
  - Box and whisker plots
  - Scatter plots

Source: (Brownlee 2016)

A recipe for practitioners

4. Preprocess data
   - Standardize numerical data (e.g. mean of 0 and standard deviation of 1).
   - Normalize numerical data (e.g. to a range of 0-1).

5. Select and compare models
   - Spot check different models for the same data

Source: (Brownlee 2016)

A recipe for practitioners

6. Evaluate your model
   - Split a dataset into training and test sets.
   - Estimate the accuracy of an algorithm using a validation approach:
     - Hold out
     - K-fold cross validation
     - Leave one out cross validation (LOOCV)
   - Compute accuracy (per class) and loss metrics
   - Analyze confusion matrix
   - Inspect prediction errors

Source: (Brownlee 2016)

A recipe for practitioners

7. Refine your model's (hyper) parameters
   - Tune the parameters of an algorithm using a grid search that you specify.
   - Tune the parameters of an algorithm using a random search.

8. Improve accuracy
   - Consider using:
     - bagging ensembles (e.g., Random Forest)
     - boosting ensembles (e.g., AdaBoost).
     - voting ensembles (by combining the predictions from multiple models together).

9. Save/export your model

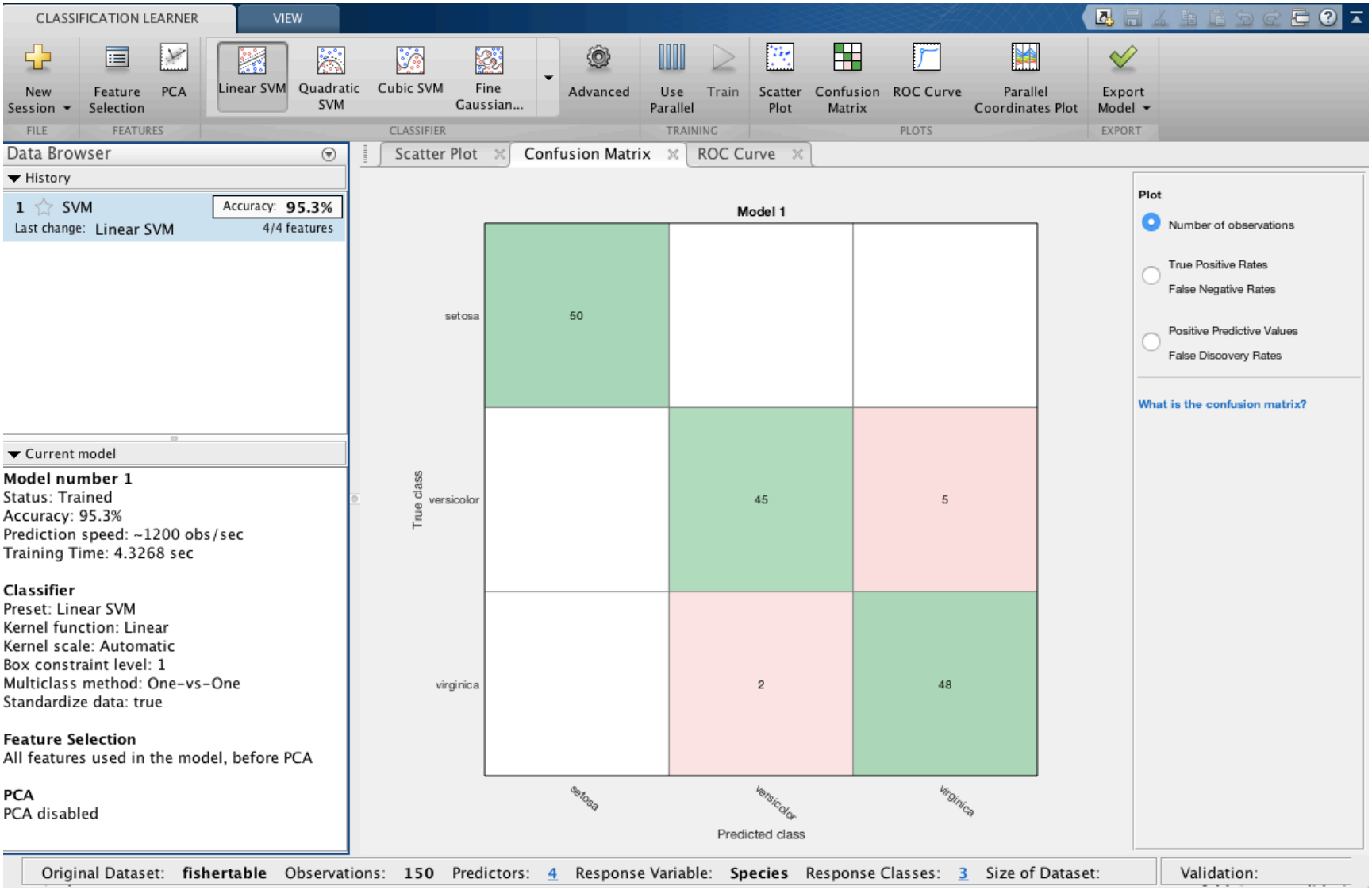
Source: (Brownlee 2016)

Machine learning using MATLAB

- **Main toolboxes**
  - Statistics and Machine Learning Toolbox
  - Neural Network Toolbox
  - Optimization Toolbox
  - Parallel Computing Toolbox
- **Auxiliary toolboxes** (for image/vision problems)
  - Image Processing Toolbox
  - Computer Vision System Toolbox
- **Third-party toolboxes**
  - MatConvNet (vlfeat.org)

Machine learning using MATLAB

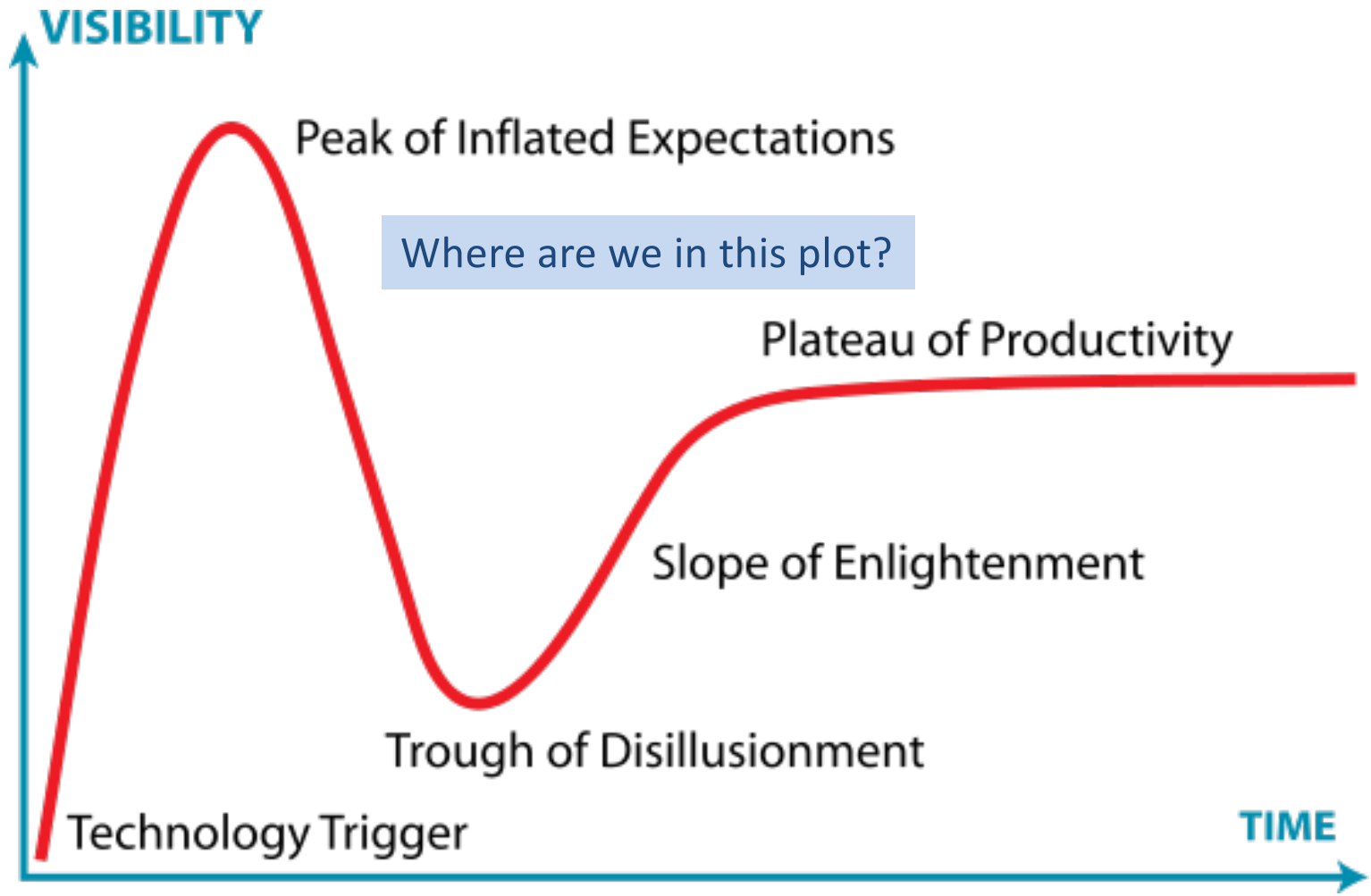
- **The Classification Learner App**



Part 3

Deep Learning

The deep learning phenomenon



Source: (Wikimedia)

Deep learning: a working definition

- Deep networks are neural networks with the following properties:
  - **more neurons** than previous neural networks
  - **more complex ways of connecting layers**
  - explosion in the amount of **computing power** available to train
  - **automatic feature learning**

Source: (Gibson&Patterson 2016)

Deep learning: the basics

- Another key driving feature of Deep Learning is that it is **able to automatically learn features** (as opposed to hand engineer features) **from data** in domain-agnostic fashion.
  - These capabilities of Deep Learning are driving many of the new technology applications and have stimulated the imagination of many lay people.
  - By itself, however, Deep Learning exhibits no higher-level functions such as “automatically understanding the most interesting question to ask a dataset”, let alone any type of sentient operation.

Source: (Gibson&Patterson 2016)

Deep learning: another definition

- Deep learning refers to neural networks with **large number of parameters and layers** in one of **four fundamental network architectures**:
  - Unsupervised Pre-Trained Networks
  - Convolutional Neural Networks
  - Recurrent Neural Networks
  - Recursive Neural Networks

Source: (Gibson&Patterson 2016)

Deep Learning = the entire machine is trainable

Traditional Pattern Recognition: Fixed/Handcrafted Feature Extractor

Feature Extractor → Trainable Classifier

Mainstream Modern Pattern Recognition: Unsupervised mid-level features

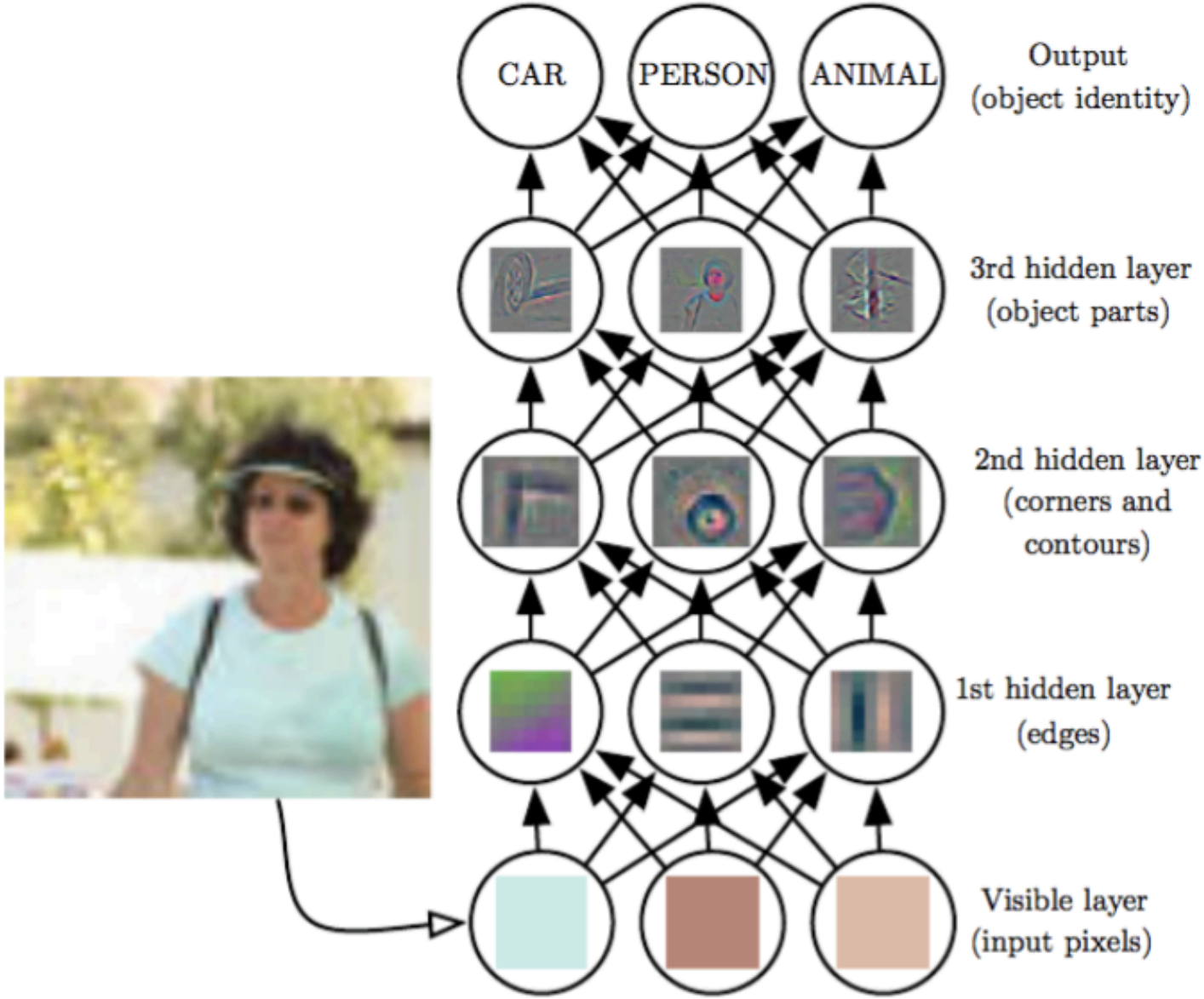
Feature Extractor → Mid-Level Features → Trainable Classifier

Deep Learning: Representations are hierarchical and trained

Low-Level Features → Mid-Level Features → High-Level Features → Trainable Classifier

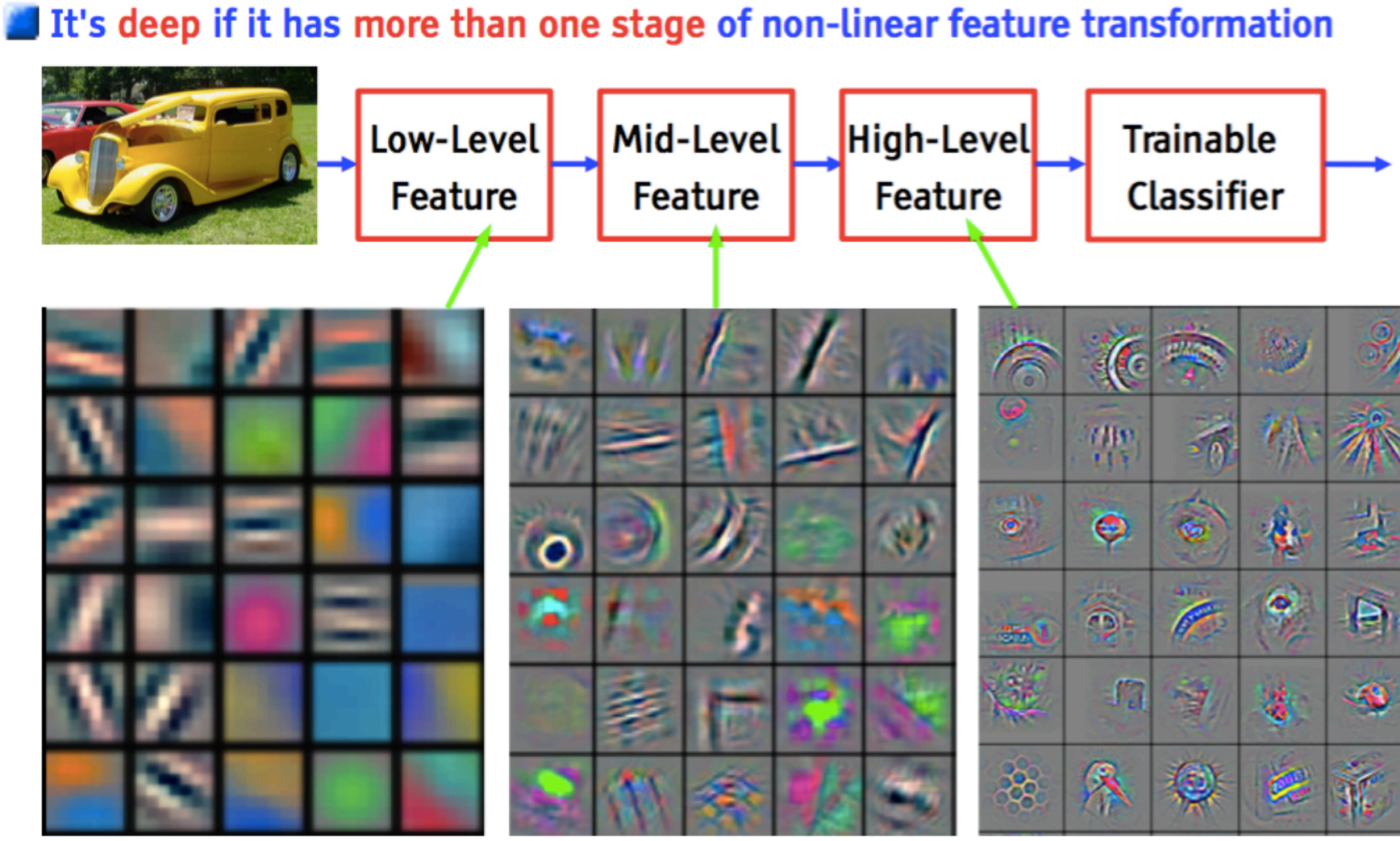
Source: (LeCun 2016)

Deep Learning: learning hierarchical representations



Source: (Goodfellow et al. 2016)

Deep Learning: learning hierarchical representations



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Source: (LeCun 2016)

Part 4

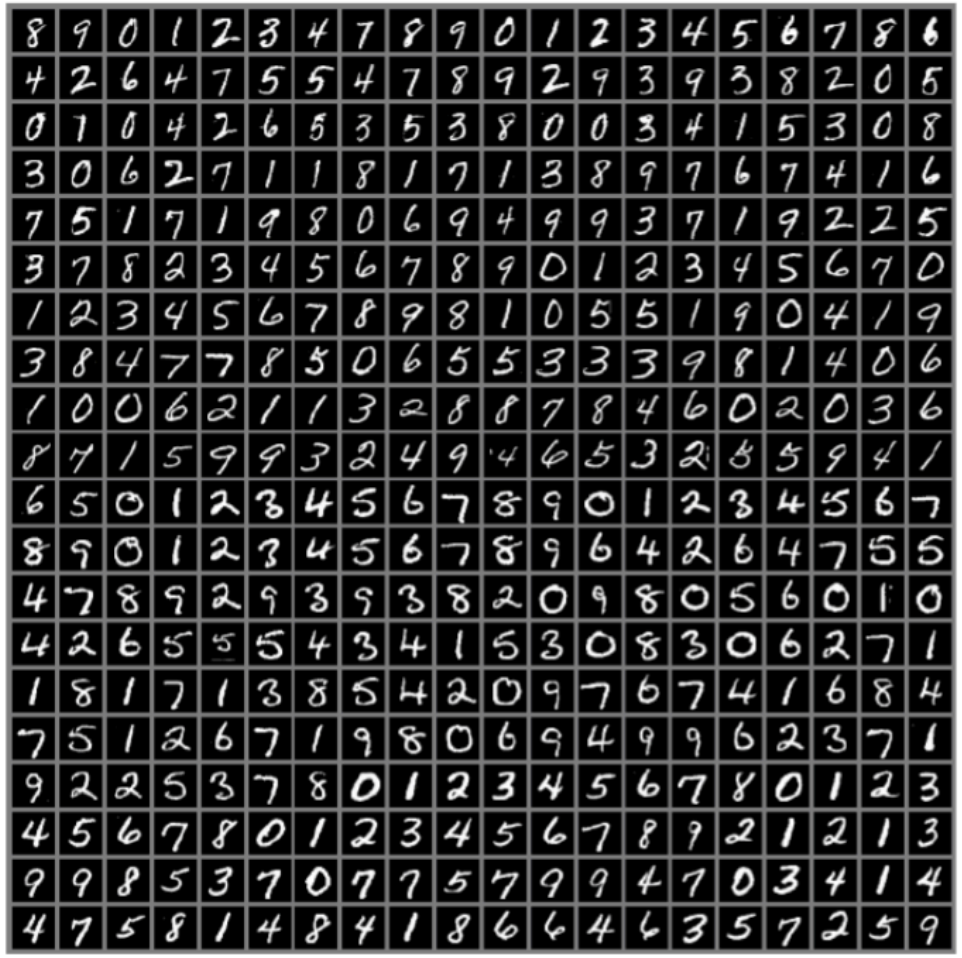
Deep Learning in computer vision and related areas

Challenges and data sets

- MNIST: http://yann.lecun.com/exdb/mnist/
- CIFAR-10 and CIFAR-100: https://www.cs.toronto.edu/~kriz/cifar.html
- ImageNet: http://image-net.org/
- Street View House Numbers (SVHN): http://ufldl.stanford.edu/housenumbers/
- STL-10: https://cs.stanford.edu/~acoates/stl10/
- Many others...
  - See http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html and http://deeplearning.net/datasets/

Datasets

- MNIST dataset
  - Scans of handwritten digits and associated labels describing which digit 0–9 is contained in each image.
  - One of the simplest and most widely used tests in deep learning research.
  - Geoffrey Hinton has described it as “the drosophila of machine learning,” meaning that it allows machine learning researchers to study their algorithms in controlled laboratory conditions, much as biologists often study fruit flies.



Source: (Goodfellow et al. 2016)

Deep learning: examples

- Deep networks perform the same modeling functions as other machine learning models (regression, classification) but have also been shown to be good at tasks such as:
  - generative modeling — generating art — generating text
  - speech recognition technology
  - image recognition technology

Source: (Gibson&Patterson 2016)

Examples

- Image caption / semantic segmentation using ConvNets

[Farabet et al. ICML 2011]

[Farabet et al. PAMI 2013]

A man riding skis on a snow covered ski slope.

NP: a man, skis, the snow, a person, a woman, a snow covered slope, a slope, a snowboard, a skier, man.
VP: wearing, riding, holding, standing on, skiing down.
PP: on, in, of, with, down.

A man wearing skis on the snow.

A man is doing skateboard tricks on a ramp.

NP: a skateboard, a man, a trick, his skateboard, the air, a skateboarder, a ramp, a skate board, a person, a woman.
VP: doing, riding, is doing, performing, flying through.
PP: on, of, in, at, with.

A man riding a skateboard on a ramp.

The girl with blue hair stands under the umbrella.

NP: a woman, an umbrella, a man, a person, a girl, umbrellas, that, a little girl, a cell phone.
VP: holding, wearing, is holding, holds, carrying.
PP: with, on, of, in, under.

A woman is holding an umbrella.

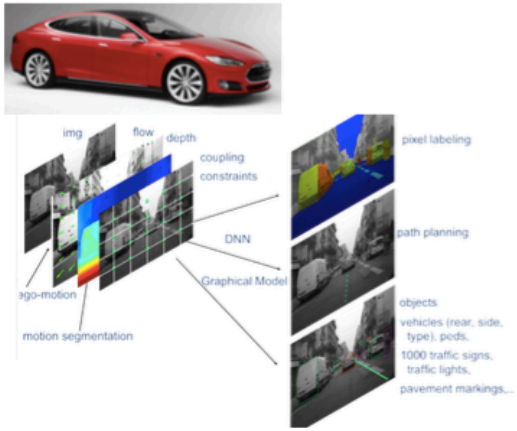
[Lebret, Pinheiro, Collobert 2015] [Kulkarni 11] [Mitchell 12] [Vinyals 14] [Mao 14] [Karpathy 14] [Donahue 14]...

Source: (LeCun 2016)

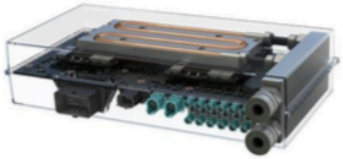
Examples

- Self-driving cars using ConvNets

MobilEye



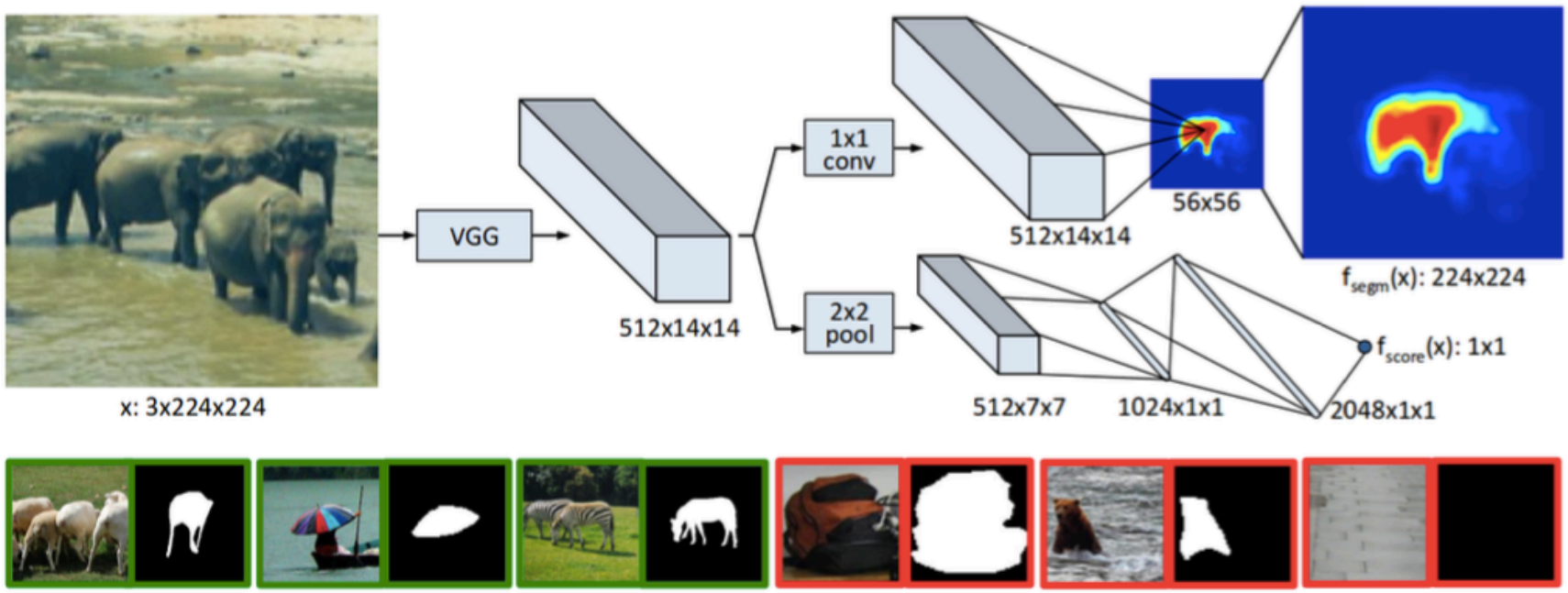
NVIDIA



Source: (LeCun 2016)

Examples

- DeepMask and SharpMask: object detection and segmentation
  - [Pinheiro, Collobert, Dollar ICCV 2015]
    - ConvNet produces object masks and categories



Source: (LeCun 2016)

Examples

- Mixing visual content and artistic style

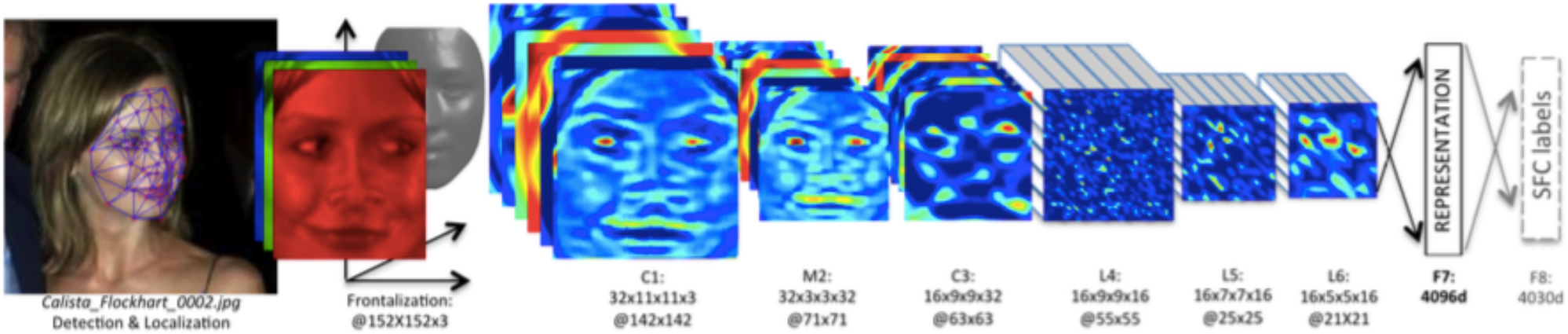
Source: (Gatys. Ecker, Bethge 2015)

Examples

DeepFace: Closing the Gap to Human-Level Performance in Face Verification

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf

Conference on Computer Vision and Pattern Recognition (CVPR) · June 24, 2014



- **Training Set**: 4.4 million labeled faces from 4,030 people (each with 800 to 1200 faces) from *Facebook*, the Social Face Classification (*SFC*) dataset.
- **Test Set**: Labeled Faces in the Wild (LFW)
- **Accuracy**: ~ 97 %

Research at Facebook

Pushing the limits…

How good are current approaches? Maybe TOO good!

LILY HAY NEWMAN SECURITY 09.12.16 11:54 AM

AI CAN RECOGNIZE YOUR FACE EVEN IF YOU'RE PIXELATED

96% Accuracy!

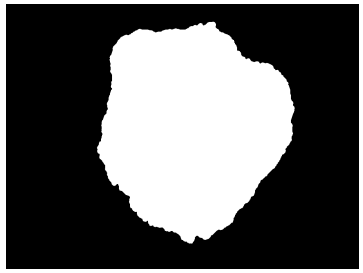
58% Accuracy!

Our work

- AI
  - OCR using a “learning from examples” approach (Philips Research Labs, 1988-89)
- ML
  - Object detection and segmentation using semi-supervised learning (Ask’nSeek and Click’n’Cut)
- DL
  - Overlapped fingerprint separation (*ongoing*)
  - Skin lesion segmentation and classification (*ongoing*)

Our work

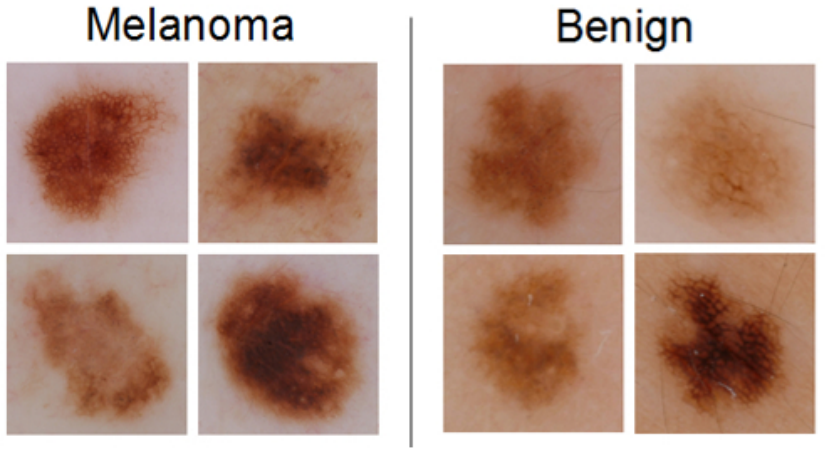
- Skin lesion segmentation
  - Skin lesion measurements
  - Skin lesion counting



- Skin lesion classification
  - Imaging-based skin disease diagnosis systems
  - Goals
    - *Achieve (or improve upon) state of the art results for skin lesion segmentation*
    - *Achieve (or improve upon) state of the art results for skin lesion classification*

Our work

- Skin lesion classification
  - Convolutional Network Architecture
    - VGG16 Network
  - Dataset
    - ISBI 2016 Challenge dataset for "Skin Lesion Analysis towards melanoma detection" (900 training images, 379 testing images)



- Methods
  1. Training from scratch
  2. ConvNet as a feature extractor (Transfer Learning)
  3. Fine-tuning the ConvNet
- Framework
  - Keras, a deep learning framework for Python

Our work

- Skin lesion classification
  - Results

The *13th* IASTED International Conference on
Biomedical Engineering
***~BioMed* 2017~**

February 20 – 22, 2017
Innsbruck, Austria

Table 1
Model evaluation of each method

Method	Train		Test	
	Loss	Accuracy (%)	Loss	Accuracy (%)
1	0.4611	81.14	0.4580	81.74
2	**0.1061**	**96.20**	0.5442	75.99
3	0.3585	84.26	**0.3994**	**85.71**

Note: ISBI 2016 Challenge winner achieved **85.5%** accuracy.

+0.21% improved

Part 5

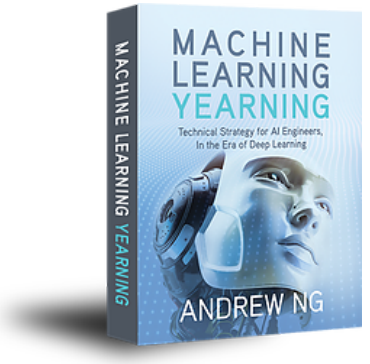
Concluding remarks

Words of advice

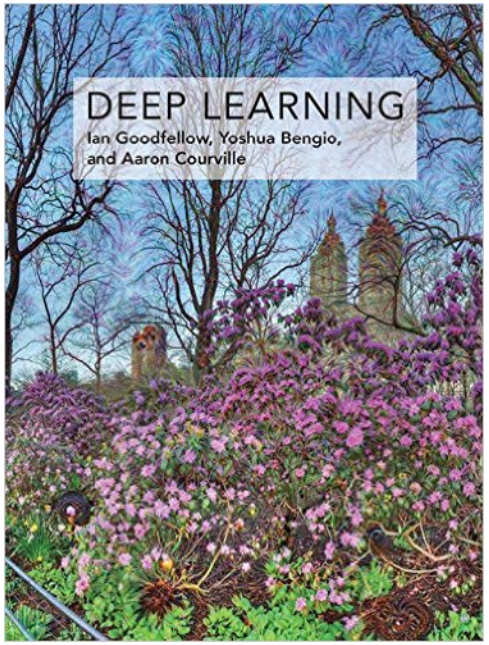
- Start small
- Select the best tools for the job
- Don’t reinvent the wheel!
- Benchmark your solutions
- Learn, learn, and learn some more!

Recommended resources

- ML
  - Andrew Ng's book (draft): http://www.mlyearning.org/
  - Machine Learning Mastery: http://machinelearningmastery.com/



- DL
  - Book: http://www.deeplearningbook.org/
  - Github:
    - https://github.com/ChristosChristofidis/awesome-deep-learning
    - https://github.com/kjw0612/awesome-deep-vision
  - MATLAB: https://www.mathworks.com/campaigns/products/offer/deep-learning-conf.html



Let's get to work!

FAU FLORIDA ATLANTIC UNIVERSITY

Oge Marques, Ph.D.
Professor

Computer & Electrical Engineering and Computer Science (CEECS)
777 Glades Road
Boca Raton, FL 33431-0991
tel: 561.297.3857
fax: 561.297.2800
email: omarques@fau.edu
Skype: ProfessorOge

https://www.facebook.com/ProfessorOgeMarques