INTRODUCTION TO MACHINE LEARNING

한재근 과장 | jahan@nvidia.com
유현곤 부장 | hryu@nvidia.com / 양한별 과장 | hanbyuly@nvidia.com

NVIDIA

AGENDA

Introduction to Deep Learning

Types of Machine Learning

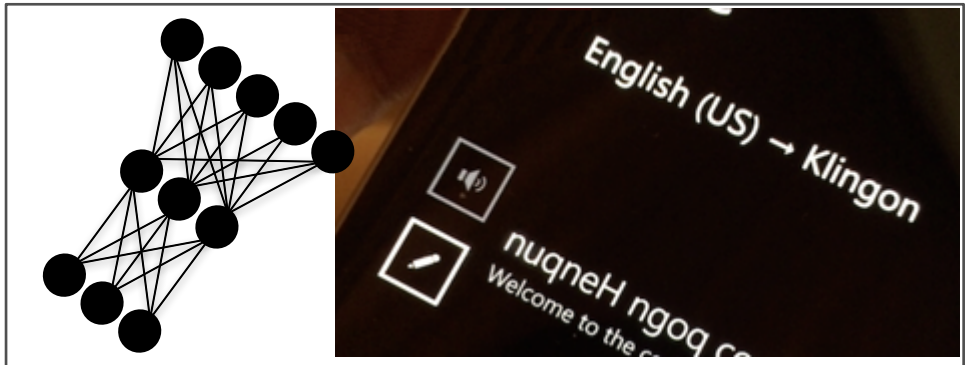
Example of Machine Learning

Feature of Machine Learning

Data features

Deep Learning - from research to technology

Image Classification, Object Detection, Localization,
Action Recognition, Scene Understanding



Speech Recognition, Speech Translation,
Natural Language Processing

Pedestrian Detection, Traffic Sign Recognition

Breast Cancer Cell Mitosis Detection,
Volumetric Brain Image Segmentation

Deep Learning success

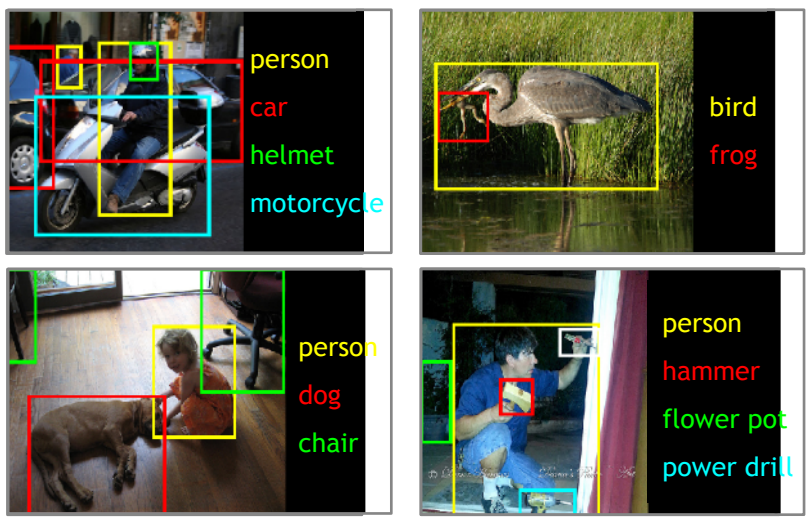
Object classification and localization in images

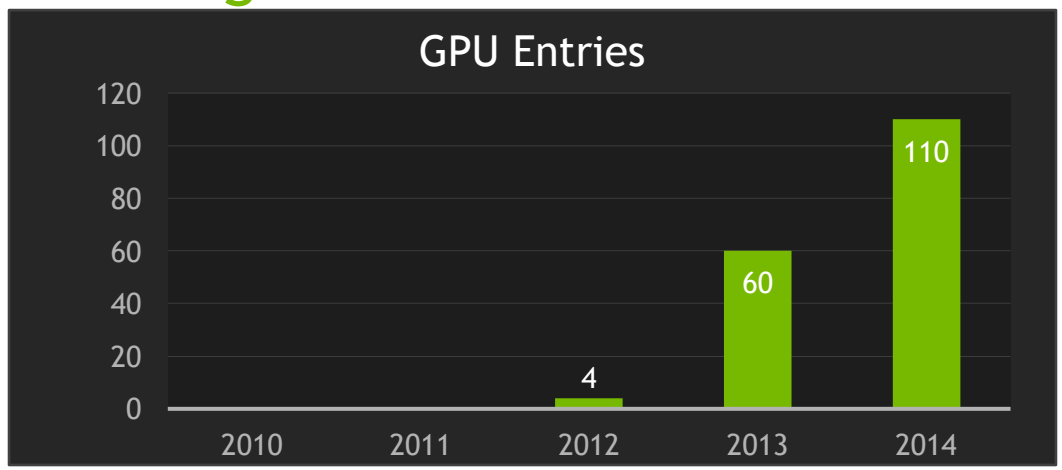
Image Recognition Challenge

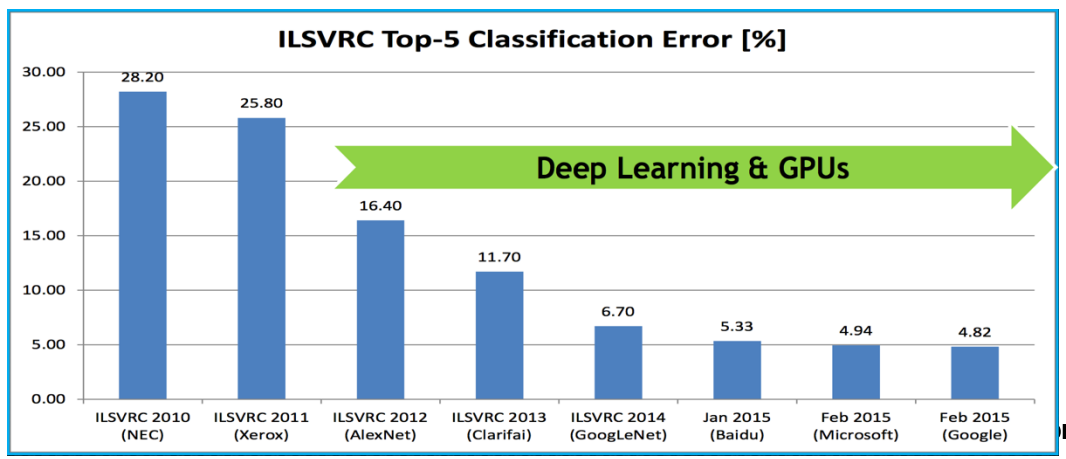
1.2M training images • 1000 object categories

Hosted by

IMAGENET

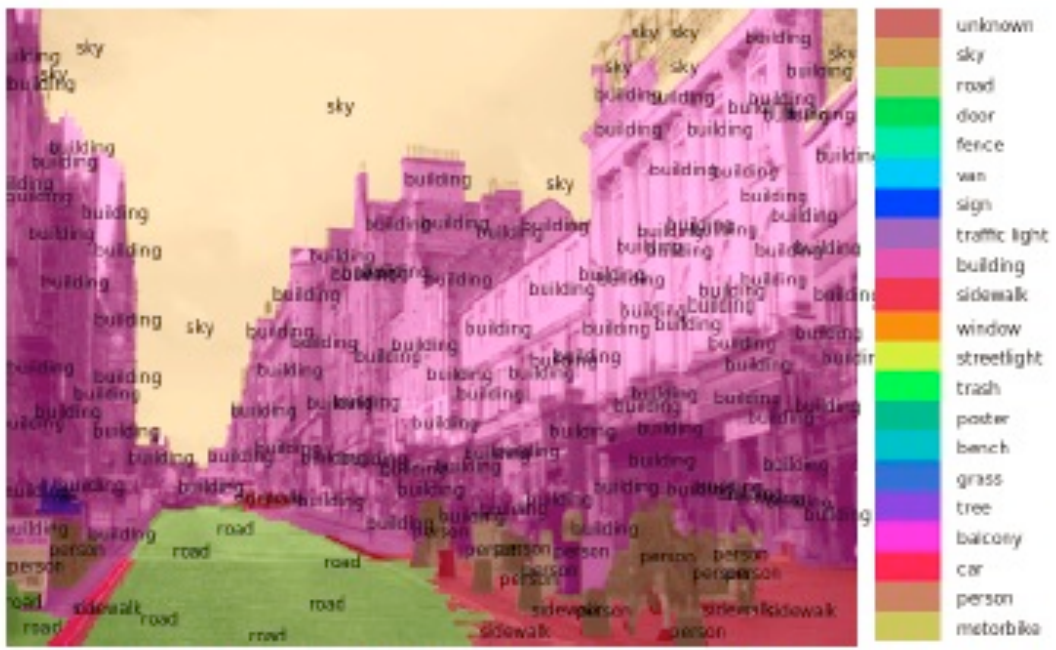






Deep Learning success

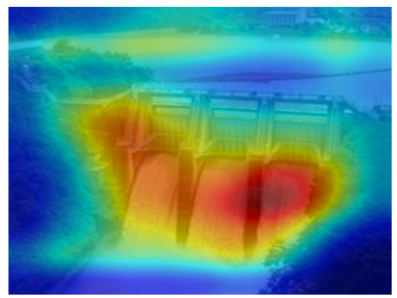
Scene segmentation and classification



Farabet, PAMI 2013

Predictions:

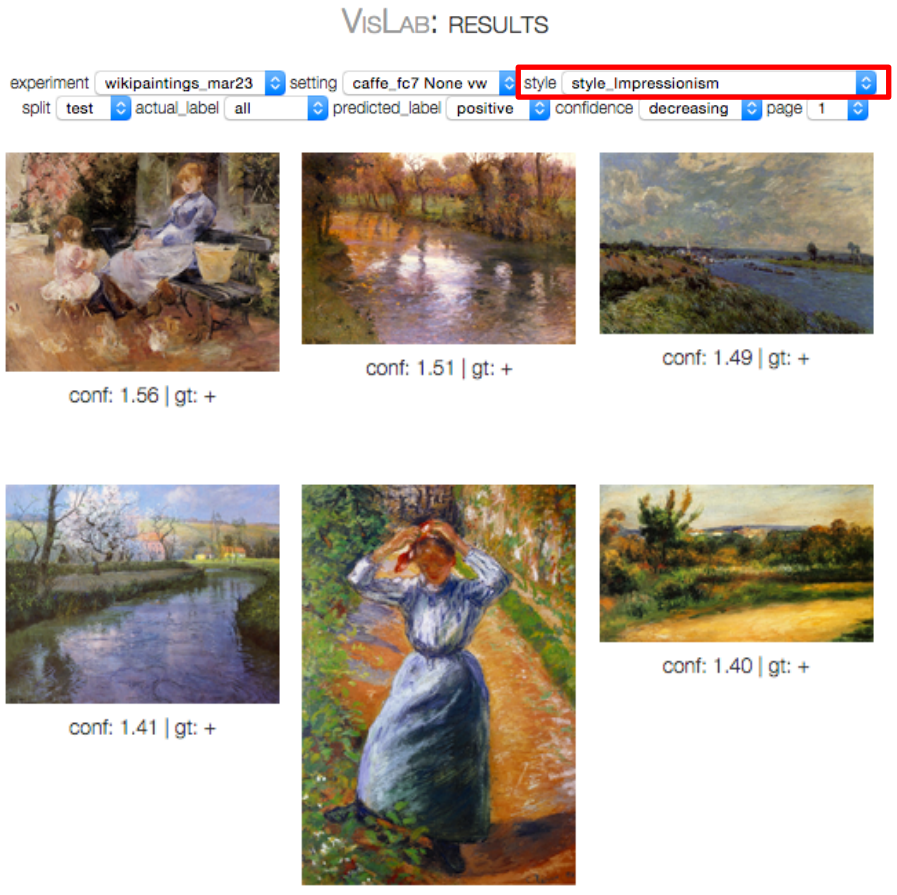
- **Type of environment:** outdoor
- **Semantic categories:** dam:1.00,
- **SUN scene attributes:** naturallight, openarea, man-made, foliage, leaves, moistdamp, runningwater, vegetation, nohorizon, shrubbery
- **Informative region for the category *dam* is:**



http://places.csail.mit.edu/ B. Zhou et al. NIPS 14

Deep Learning success

Artistic style recognition and imitation



http://demo.vislab.berkeleyvision.org/

The Deep Forger
@DeepForger
Follow

#StyleNet #NeuralArt, comission from @FinnSiegmund and style by Pablo Picasso.

Deep Learning success

Activity recognition in video

https://www.youtube.com/watch?v=qrzQ_AB1DZk Google/Stanford

Deep Learning success

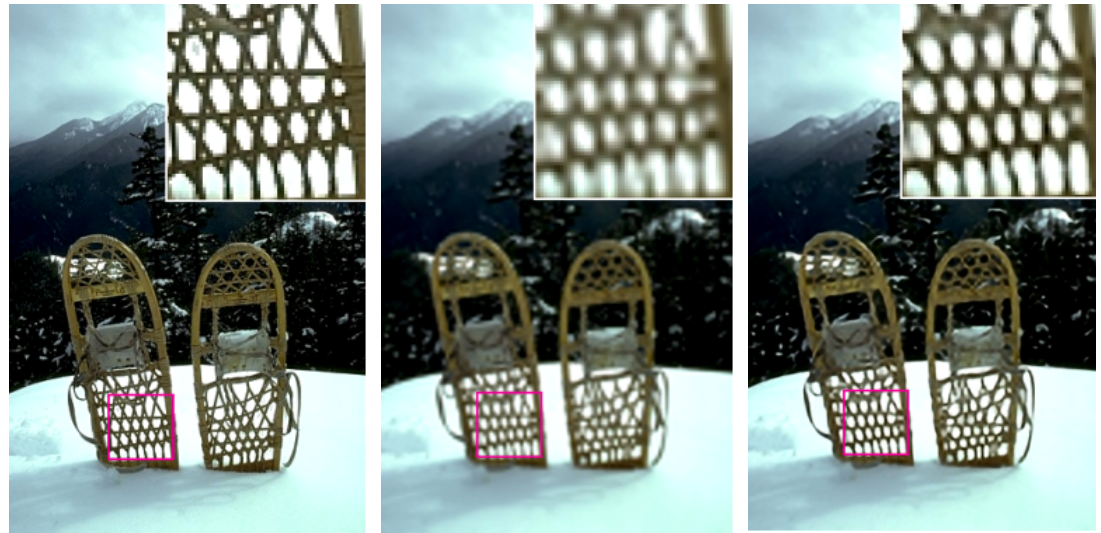
Automotive and embedded systems

https://www.youtube.com/watch?v=fmVWLr0X1Sk&feature=youtu.be

Deep Learning success

Image noise reduction and upscaling

Eigen , ICCV 2010



Dong et al, ECCV 2014

Deep Learning success

End-to-end speech recognition

Baidu system is significantly simpler than traditional speech recognition systems, which rely on laboriously engineered processing pipelines.

Deep speech does not need hand-designed components to model background noise, speaker variation etc, but instead directly learns them

System	Clean (94)	Noisy (82)	Combined (176)
Apple Dictation	14.24	43.76	26.73
Bing Speech	11.73	36.12	22.05
Google API	6.64	30.47	16.72
wit.ai	7.94	35.06	19.41
Deep Speech	**6.56**	**19.06**	**11.85**

Table 4: Results (%WER) for 5 systems evaluated on the original audio. Scores are reported *only* for utterances with predictions given by all systems. The number in parentheses next to each dataset, e.g. Clean (94), is the number of utterances scored.

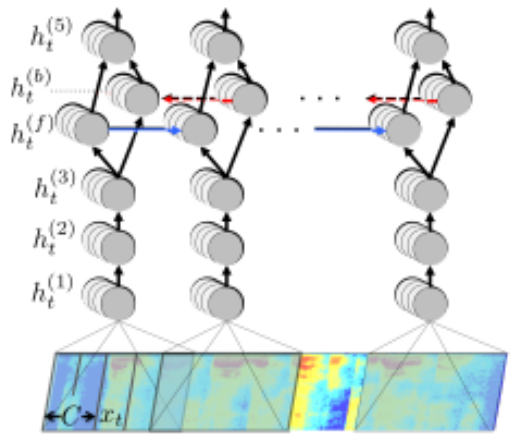


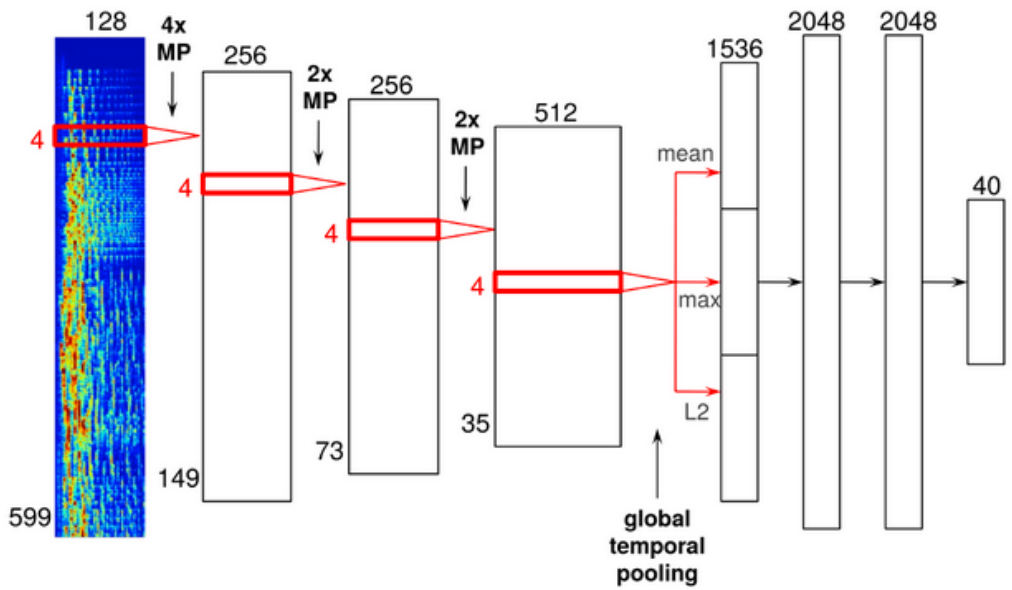
Figure 1: Structure of our RNN model and notation.

Deep Learning success

Style based music recommendation

http://benanne.github.io/2014/08/05/spotify-cnns.html

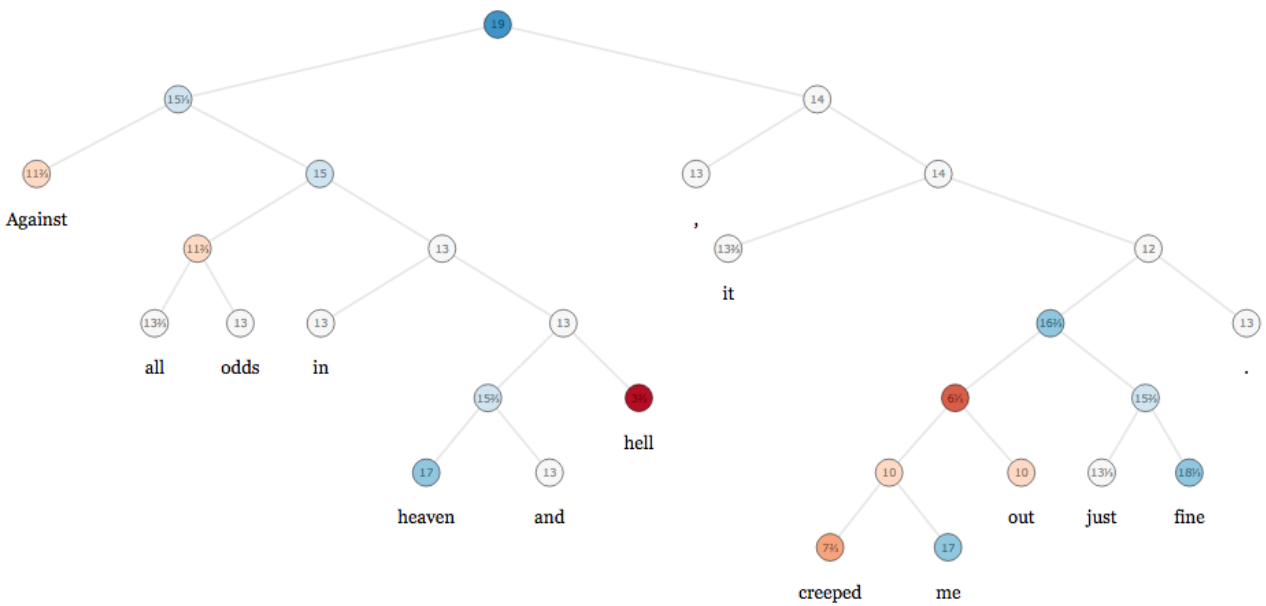
Audio spectrogram input

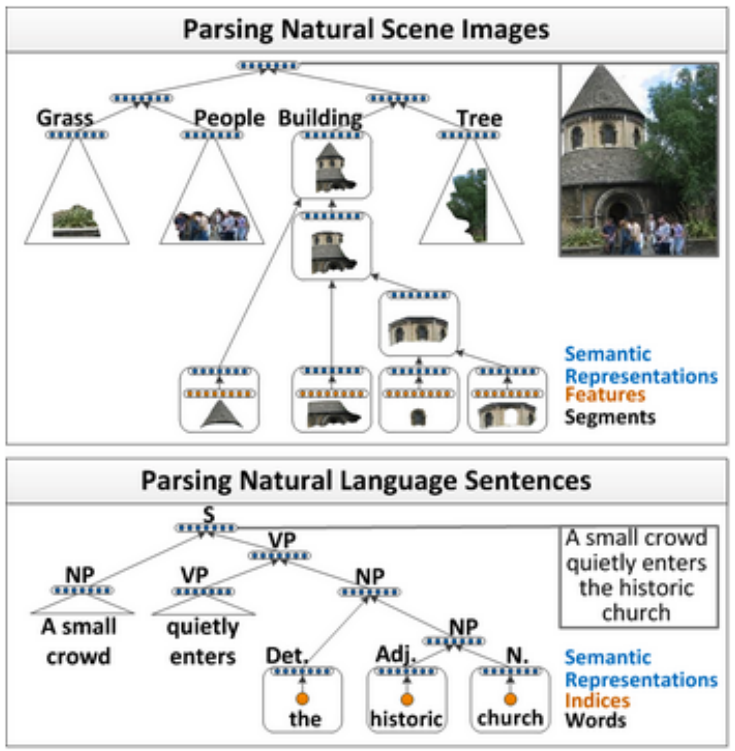


Latent factors, i.e “tags” output

Deep Learning success

Language understanding





Stanford NLP Group

Deep Learning success

Playing games

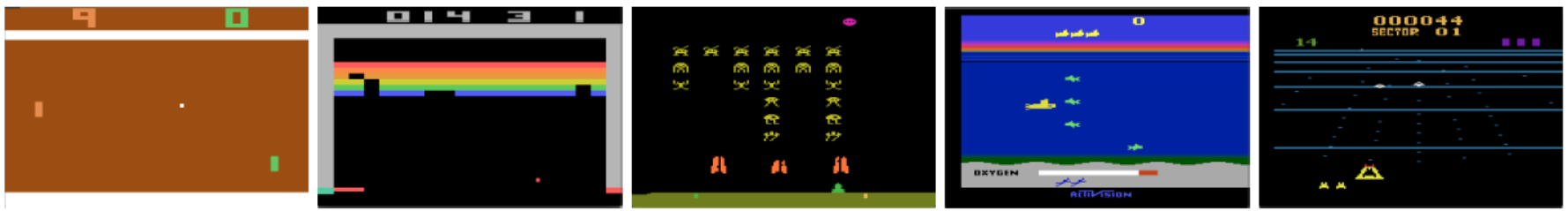
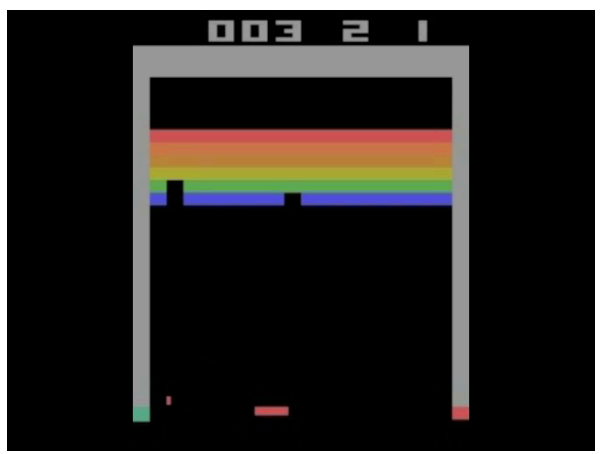


Figure 1: Screen shots from five Atari 2600 Games: (*Left-to-right*) Pong, Breakout, Space Invaders, Seaquest, Beam Rider



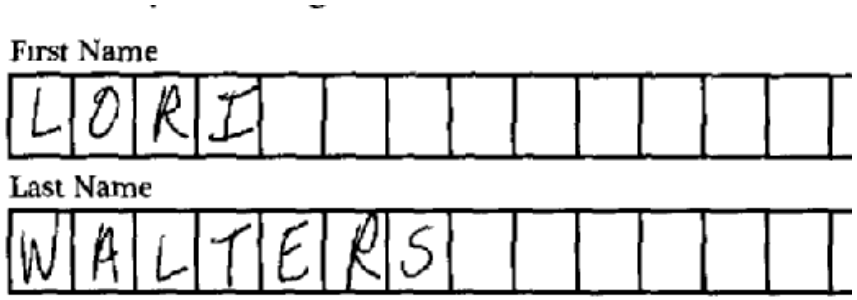
Google DeepMind

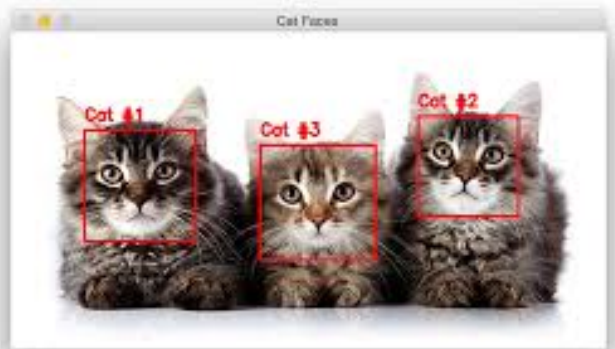
[Cookie Run]
https://www.youtube.com/watch?v=exXD6wJLJ6s

Machine Learning Fundamentals

Machine Learning

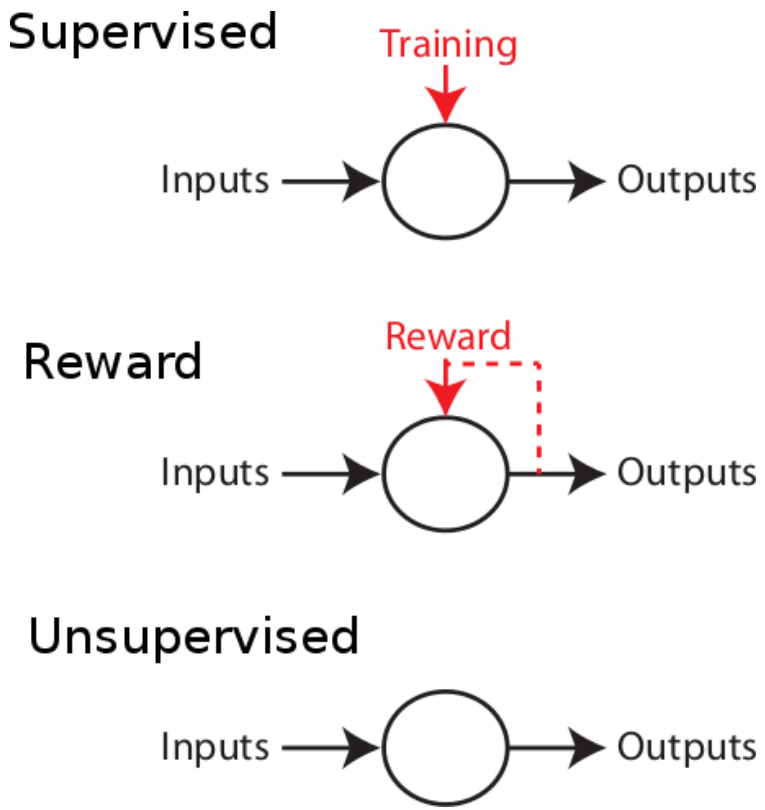
- Machine Learning is the ability to teach a computer without explicitly programming it
- Examples are used to train computers to perform tasks that would be difficult to program

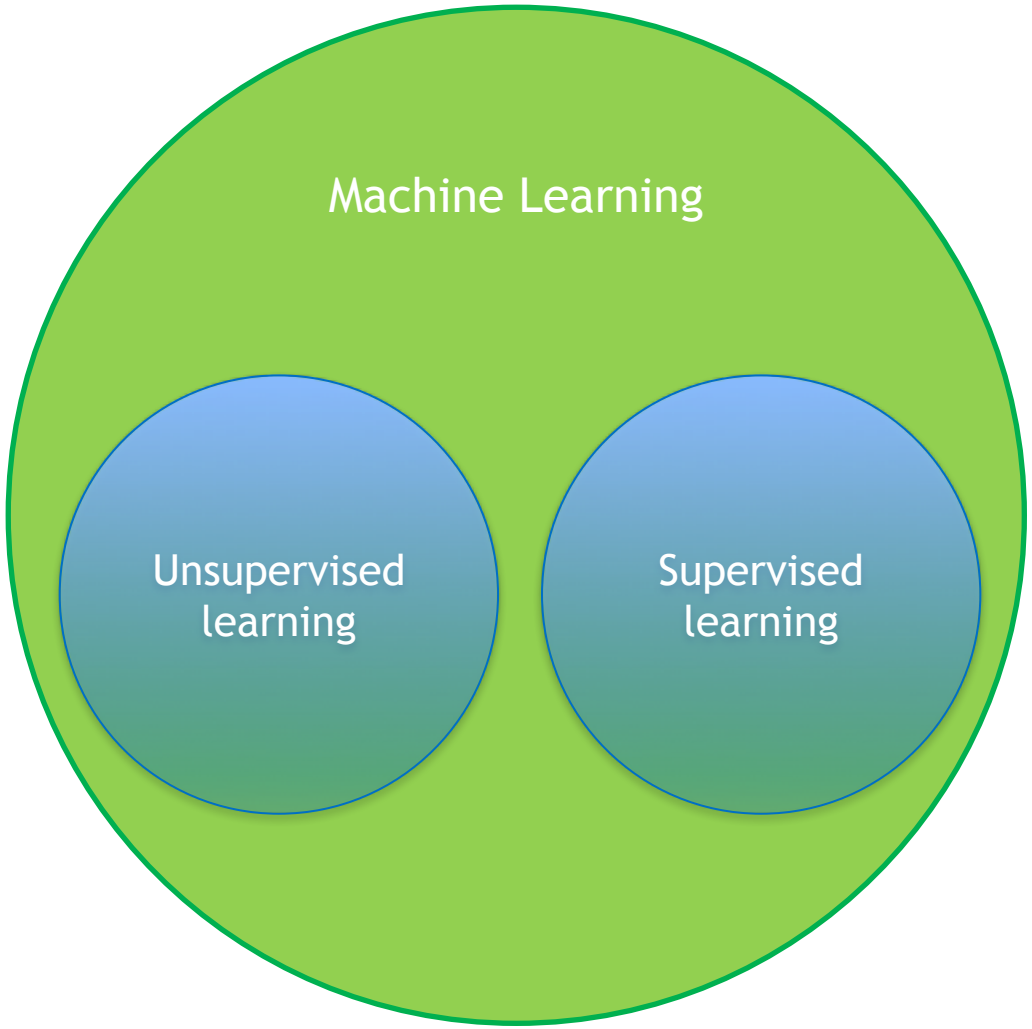




Types of machine Learning

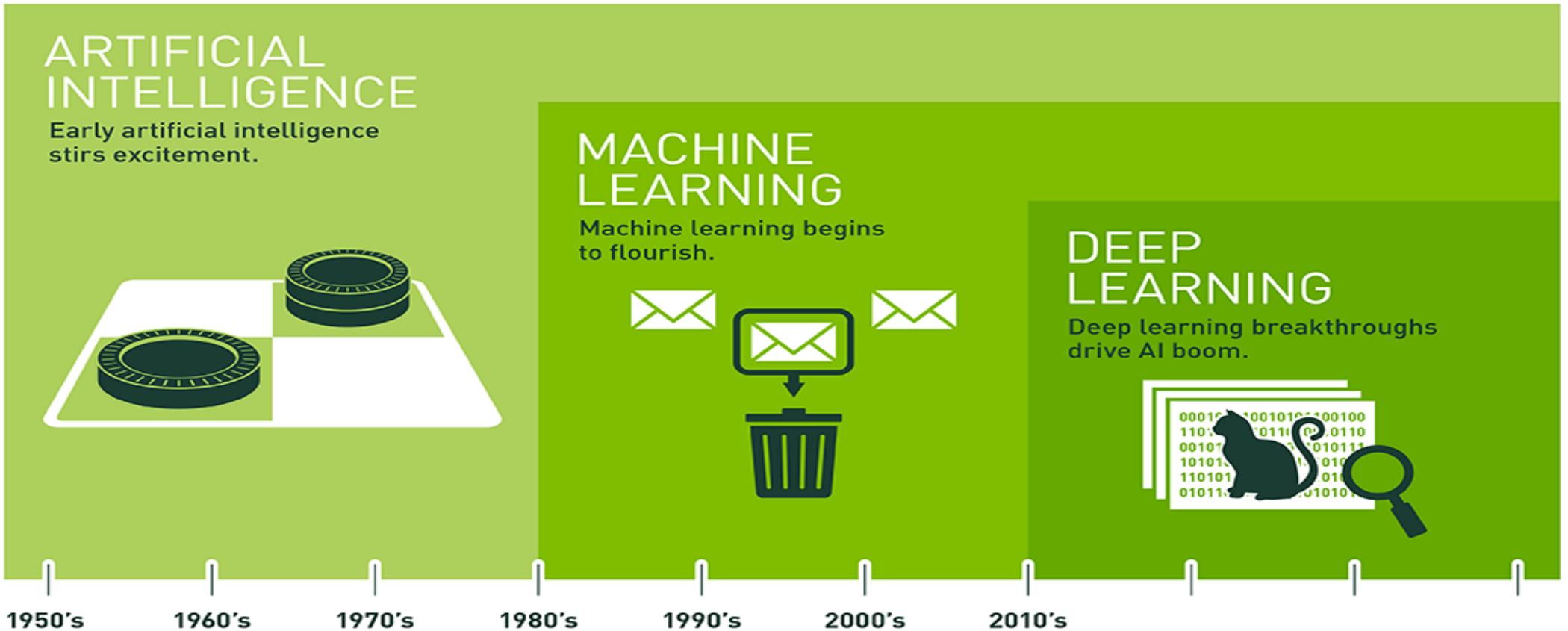
- Supervised Learning
  - Training data is labeled
  - Goal is correctly label new data
- Reinforcement Learning
  - Training data is unlabeled
  - System receives feedback for its actions
  - Goal is to perform better actions
- Unsupervised Learning
  - Training data is unlabeled
  - Goal is to categorize the observations





- **Supervised** – labelled data
  - Convolutional NN (LeCun)
  - Recurrent NN (Schmidhuber)
- **Unsupervised** – no labels
  - Deep Belief Nets/stacked RBMs (Hinton)
  - Autoencoders (Bengio, LeCun, Ng)

Capability of Machine to imitate intelligent behavior



LEARNING FROM DATA

AND SOME BUZZ WORDS

ARTIFICAL INTELLIGENCE

Knowledge & Reason

Learning

Planning

Communicating

Perceiving

MACHINE LEARNING

Learning from data

Expert systems

Handcrafted features

DEEP LEARNING

Learning from data

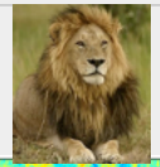
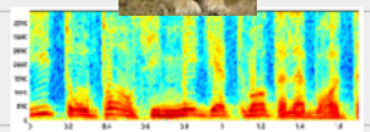
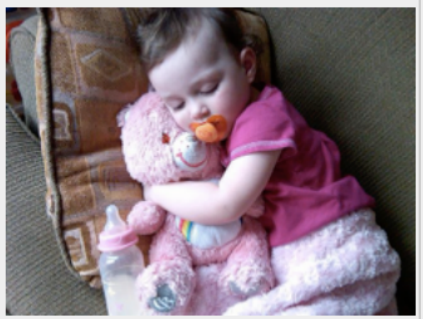
Neural networks

Computer learned features

Applications of Machine Learning

- Handwriting Recognition
  - convert written letters into digital letters
- Language Translation
  - translate spoken and or written languages (e.g. Google Translate)
- Speech Recognition
  - convert voice snippets to text (e.g. Siri, Cortana, and Alexa)
- Image Classification
  - label images with appropriate categories (e.g. Google Photos)
- Autonomous Driving
  - enable cars to drive

Input-output examples

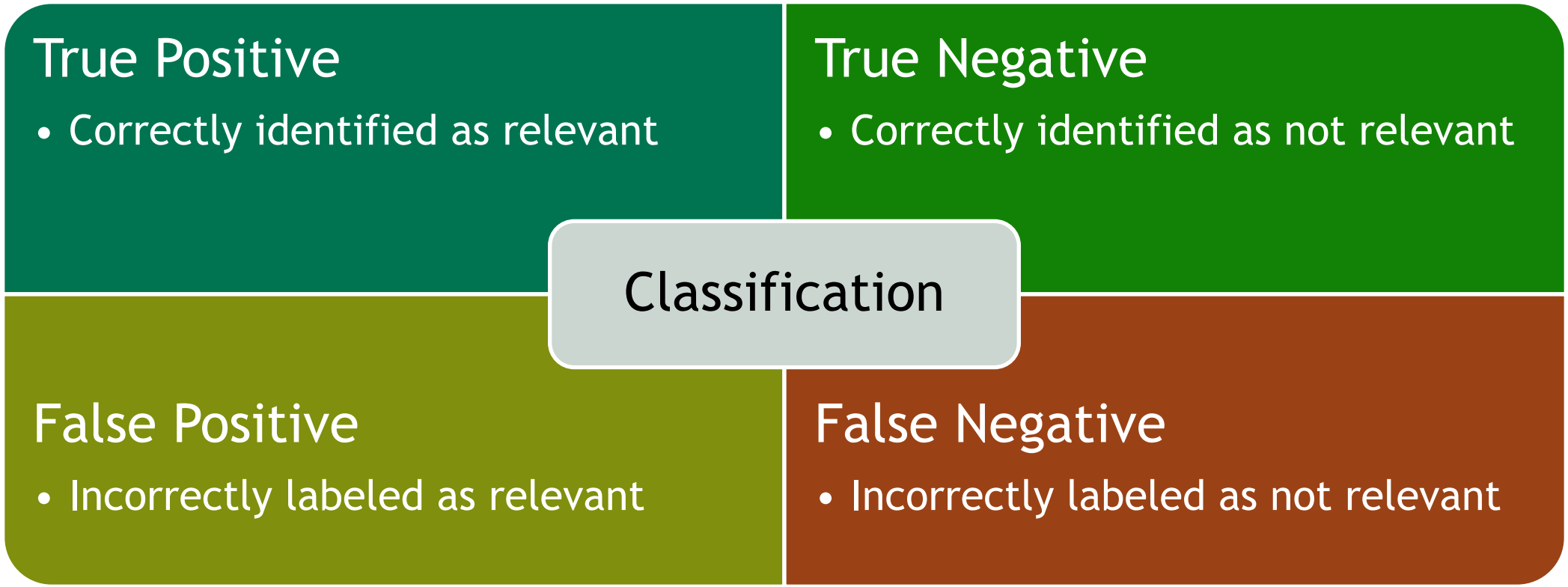
Input	Output
Pixels: 	“lion”
Audio: 	“see at tuhl res taur aun ts”
<query, doc>	P(click on doc)
“Hello, how are you?”	“Bonjour, comment allez-vous?”
Pixels: 	“A close up of a small child holding a stuffed animal”

Jeff Dean, Google, GTC 2015

Features in Machine Learning

- Features are the observations that are used to form predictions
  - For image classification, the pixels are the features
  - For voice recognition, the pitch and volume of the sound samples are the features
  - For autonomous cars, data from the cameras, range sensors, and GPS are features
- Extracting relevant features is important for building a model
  - Time of day is an irrelevant feature when classifying images
  - Time of day is relevant when classifying emails because SPAM often occurs at night
- Common Types of Features in Robotics
  - Pixels (RGB data)
  - Depth data (sonar, laser rangefinders)
  - Movement (encoder values)
  - Orientation or Acceleration (Gyroscope, Accelerometer, Compass)

Measuring Success for Classification



Example: Identify Cats

Images from the STL-10 dataset

Precision, Recall, and Accuracy

- Precision
  - Percentage of positive labels that are correct
  - Precision = (# true positives) / (# true positives + # false positives)
- Recall
  - Percentage of positive examples that are correctly labeled
  - Recall = (# true positives) / (# true positives + # false negatives)
- Accuracy
  - Percentage of correct labels
  - Accuracy = (# true positives + # true negatives) / (# of samples)

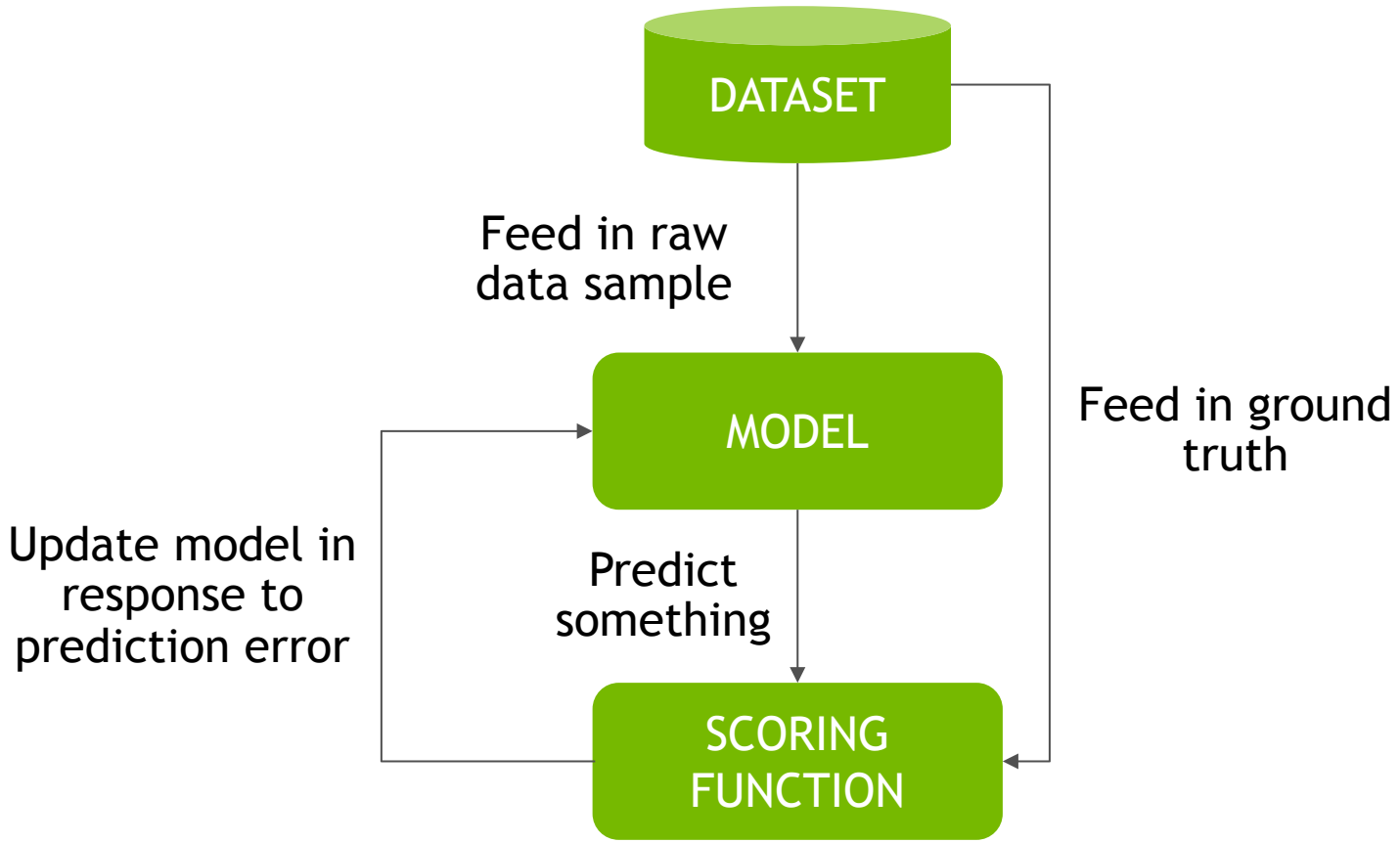
Supervised learning setup

Inputs (AKA features) - real-valued vectors of data
e.g. Image pixels, audio spectrograms, character sequences

Outputs (AKA labels) - real-valued or categorical “truth” vectors
e.g. class labels for images, audio transcription, sentiment

Training data - many samples of input-output pairs

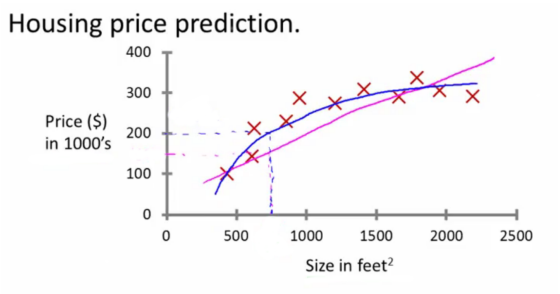
Basic machine learning workflow



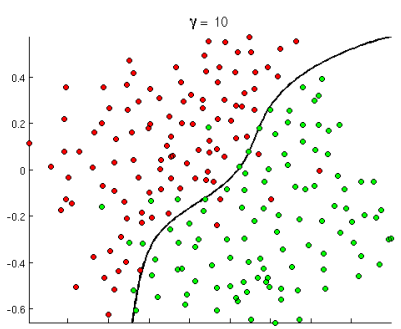
REPEAT UNTIL MODEL IS “GOOD”

Supervised Learning Objectives

- Regression – outputs are continuous/real-valued scalar or vectors
  - Example:

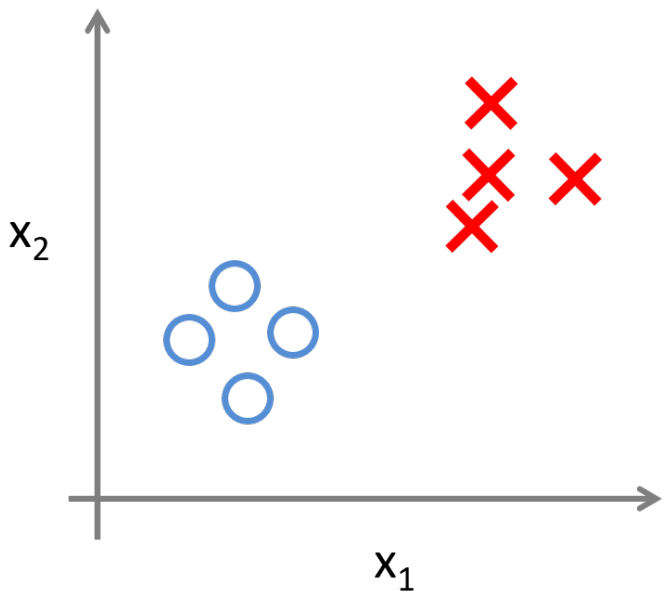


- Classification – outputs are categorical
  - Example:

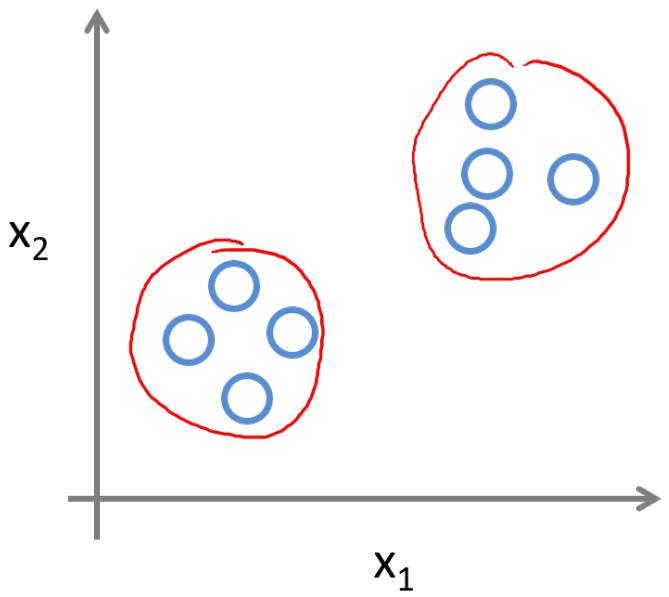


Supervised vs Unsupervised

Supervised Learning



Unsupervised Learning



Score function (AKA model)

A function that predicts the output given an input

Example: linear regression

Data

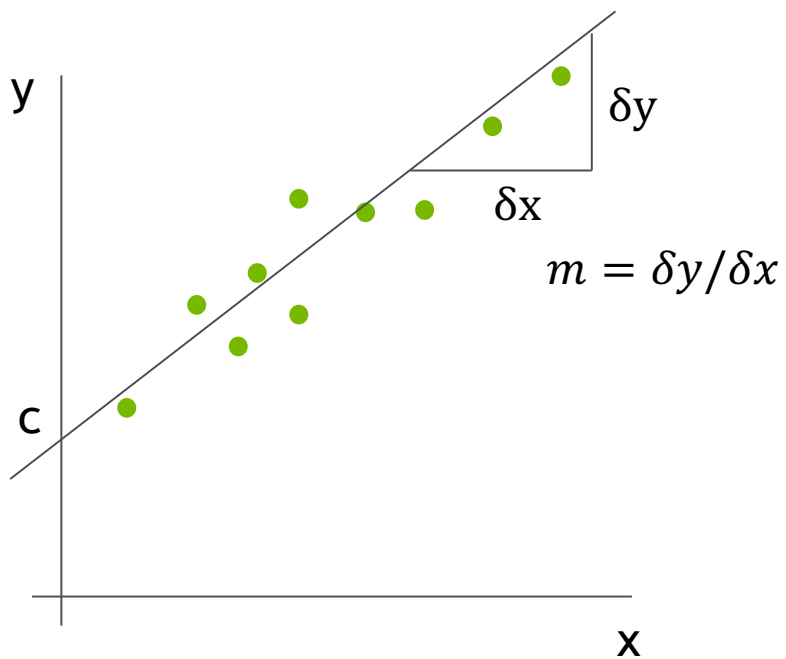
$$y_i = mx_i + c$$

Predicted output

Slope

Intercept

Together, m and c are called the **model parameters**



Loss function (AKA objective)

Measures how good a particular choice of model parameters are
This is an application dependent choice that must be made
Example: mean squared error

$$L_i = (\hat{y}_i - y_i)^2$$

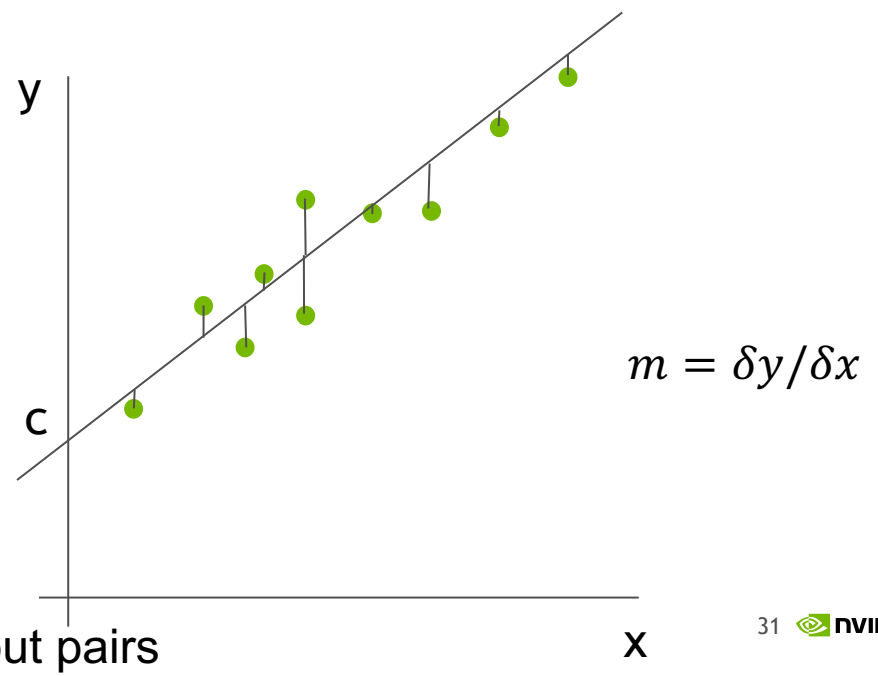
True output

Predicted output

$$L = \frac{1}{N} \sum_i L_i$$

Total loss

Sum over all input-output pairs



Bias and Variance

- Bias: expected difference between model's prediction and truth
  - Variance: how much the model differs among training sets

- Model Scenarios
  - High Bias: Model makes inaccurate predictions on training data
  - High Variance: Model does not generalize to new datasets
  - Low Bias: Model makes accurate predictions on training data
  - Low Variance: Model generalizes to new datasets

Supervised learning

In simple examples like before we can often solve for the parameters analytically

As the function that maps inputs to outputs becomes more complex we lose this ability and must learn the parameters

Training: the process of learning the model parameters that are optimal as judged by the loss function

Example: learn m and c so that the mean-squared error is minimized

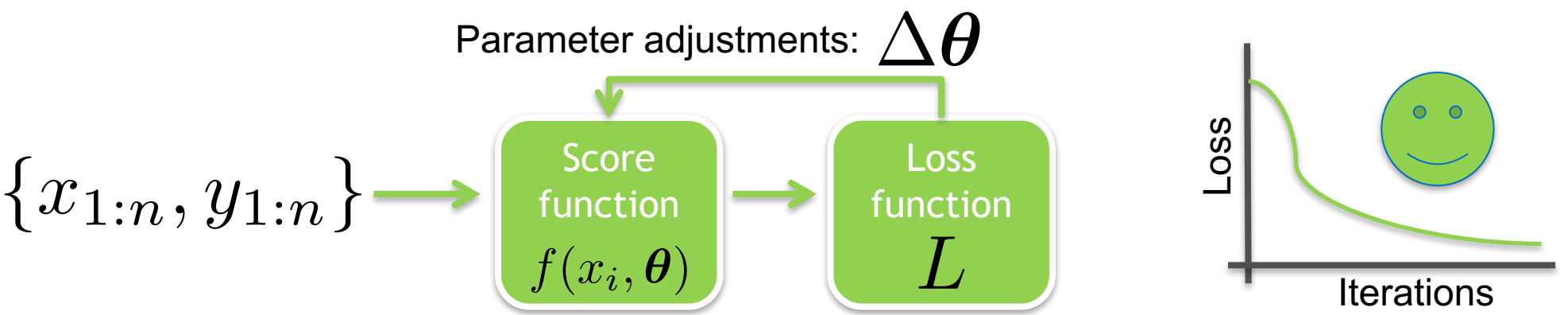
Supervised learning

How do we do this?

Repeatedly feed training data into a learning algorithm

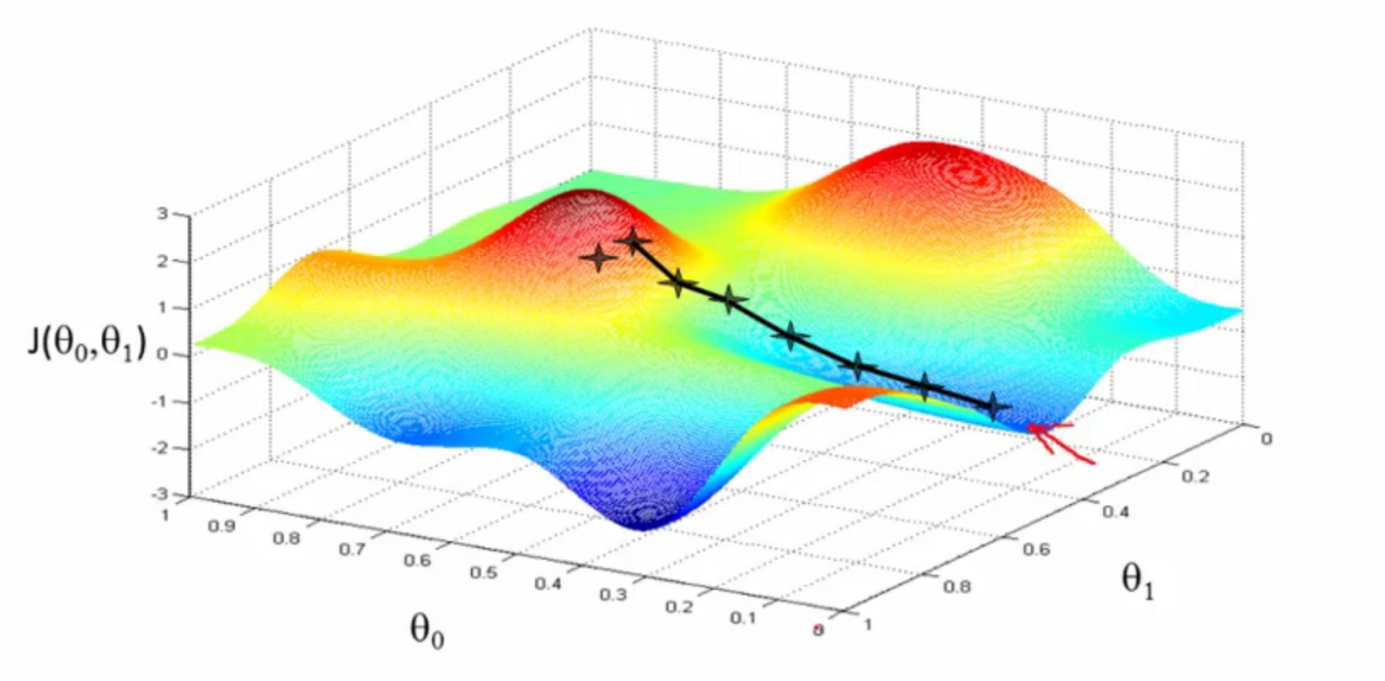
Iteratively modify the model parameters to optimize (e.g. minimize) the loss function

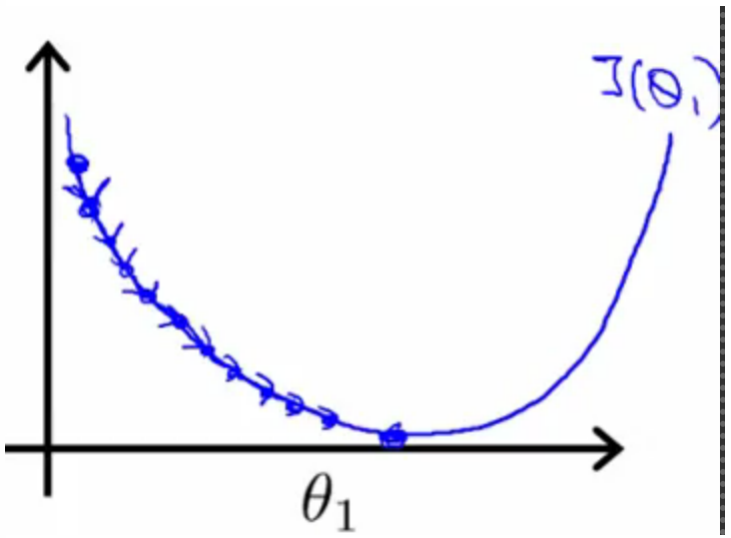
Repeat until the model is “good enough”



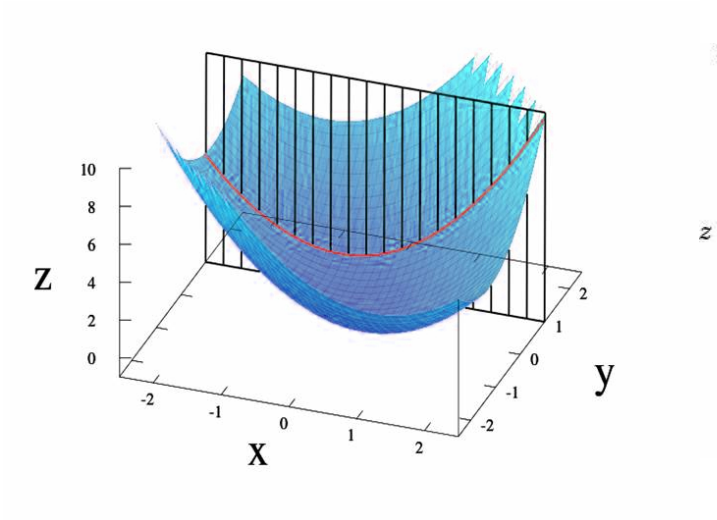
Gradient descent

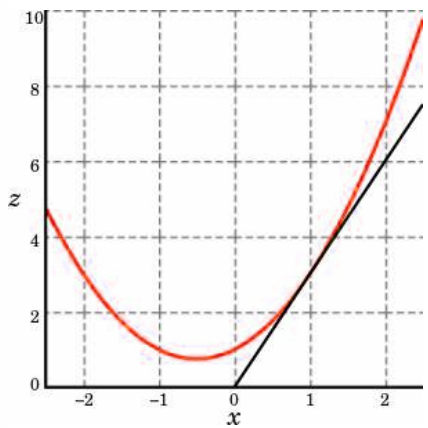
Finding the Optimal Parameters for our Hypothesis





Gradient descent - computation





Partial Derivatives

Parameter Update Algorithm

repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$

} (simultaneously update $j = 0$ and $j = 1$)

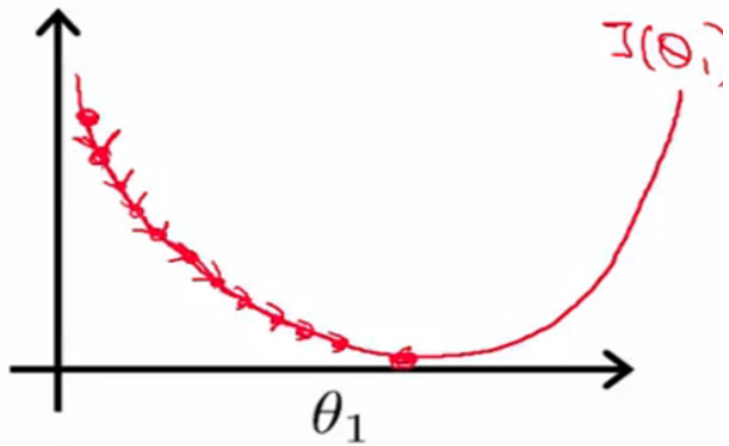
Gradient descent - Learning Rate

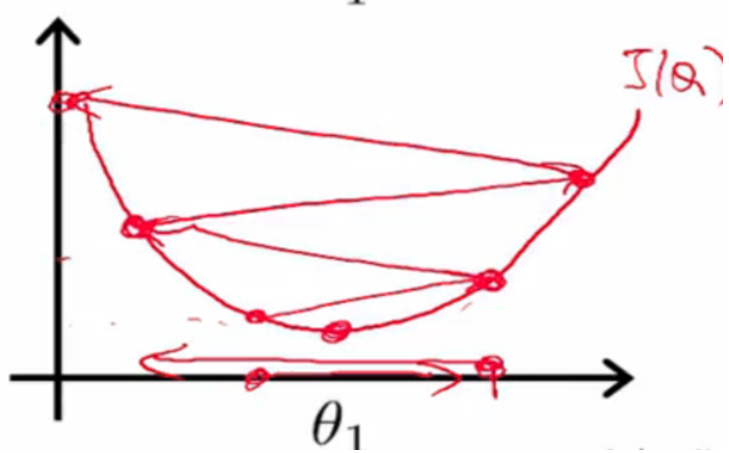
Gradient Descent Hyperparameter

$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

If α is too small, gradient descent can be slow.

If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.





Supervised learning

Why do we do this?

Given the **model** we can take previously unseen inputs and predict the corresponding output. We call this **testing** or **deployment**.

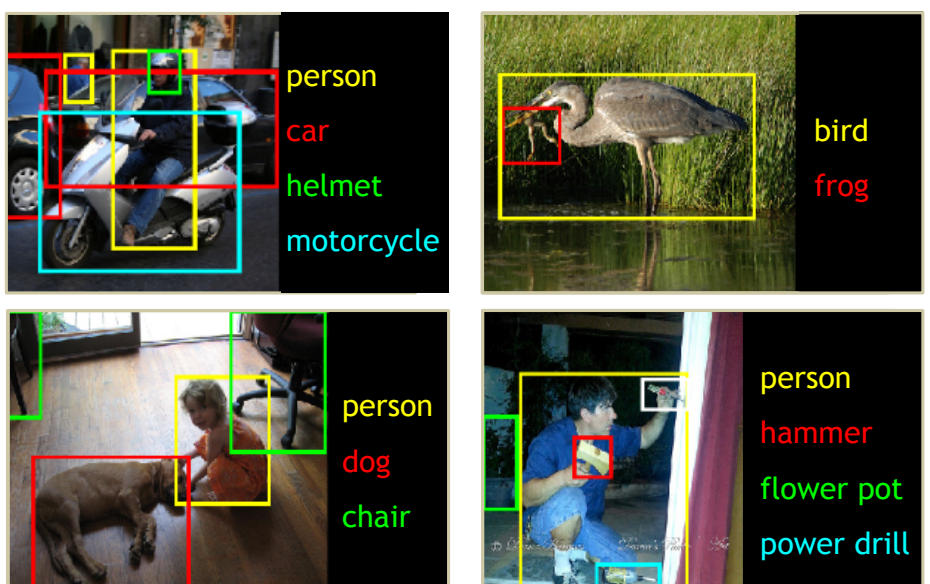
Example: ImageNet

Image recognition challenge

1.2M training images • 1000 object categories

Hosted by

IMAGENET

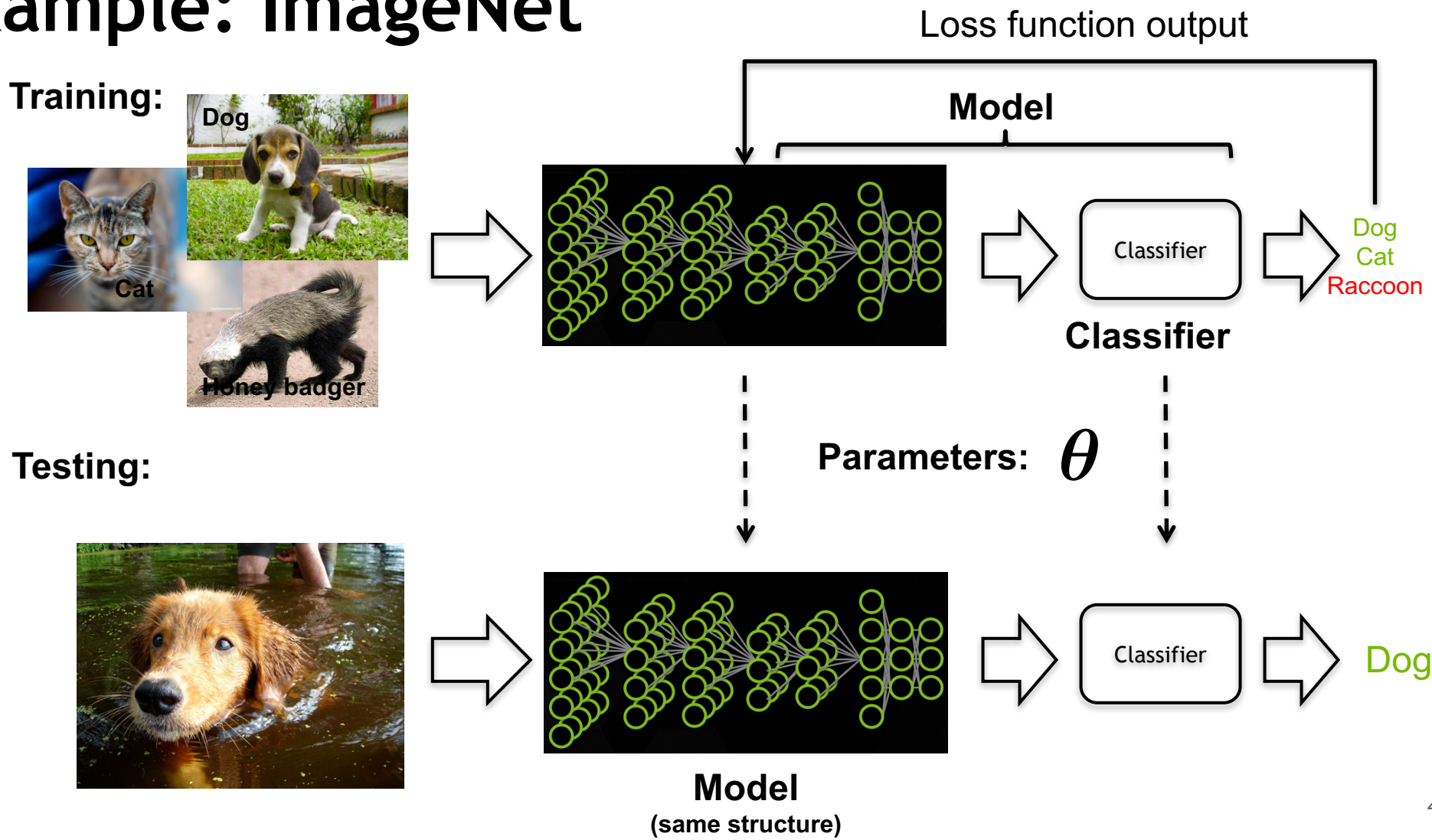


Inputs: RGB images

Outputs:

- Object labels
- Locations of objects

Example: ImageNet



Example: ImageNet

The model: convolutional neural network converting pixels to low-dimensional feature vector

Score function: maps model output to vector of confidences that each object class is in the image

Loss function: difference between object class confidence vector and vector of true object classes in image

Training problems

Two major problems

Underfitting: model is bad at it's objective for all data

Overfitting: model is really good at the objective for the training data but bad on the testing data

First line of defense:

Break off a **validation** dataset from the training data, e.g. 25%

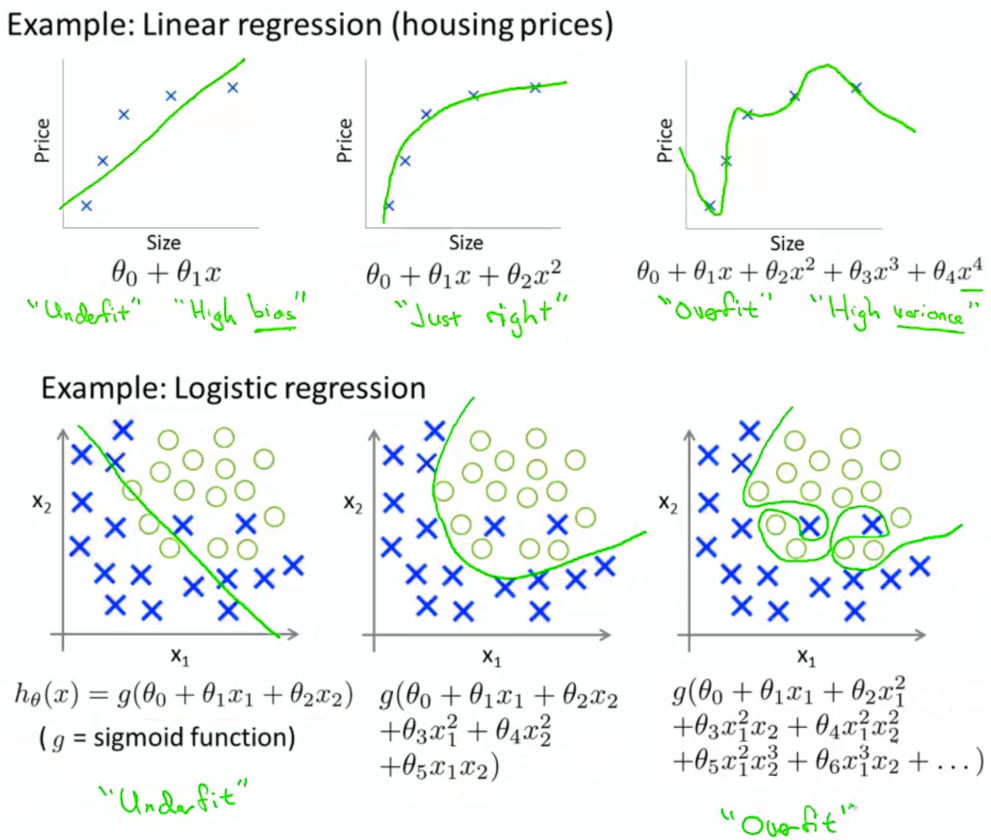
Use it during training to check model performance on unseen data

Training problems

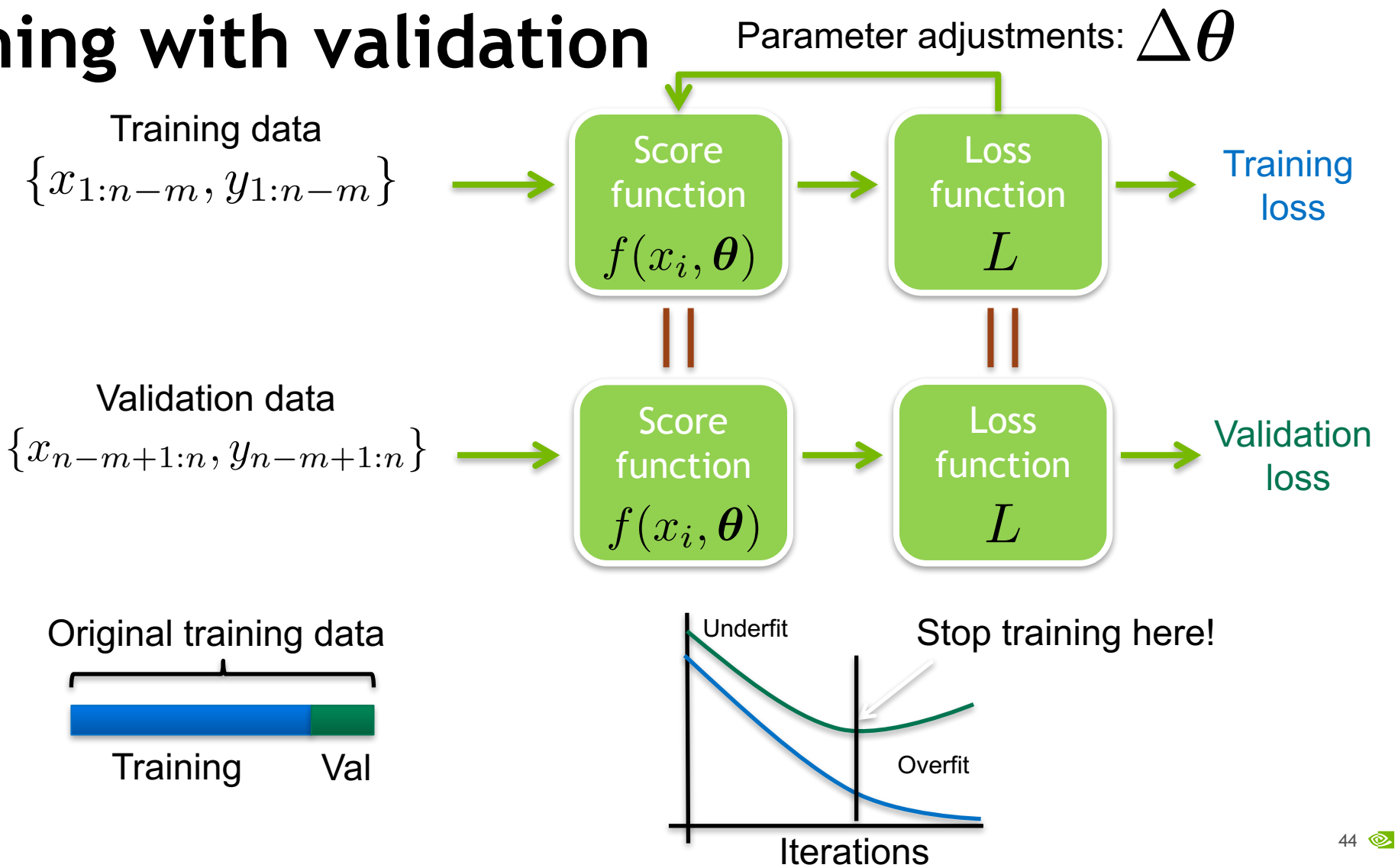
Underfit / Overfit

Low loss value may not the best

More Validation is needed

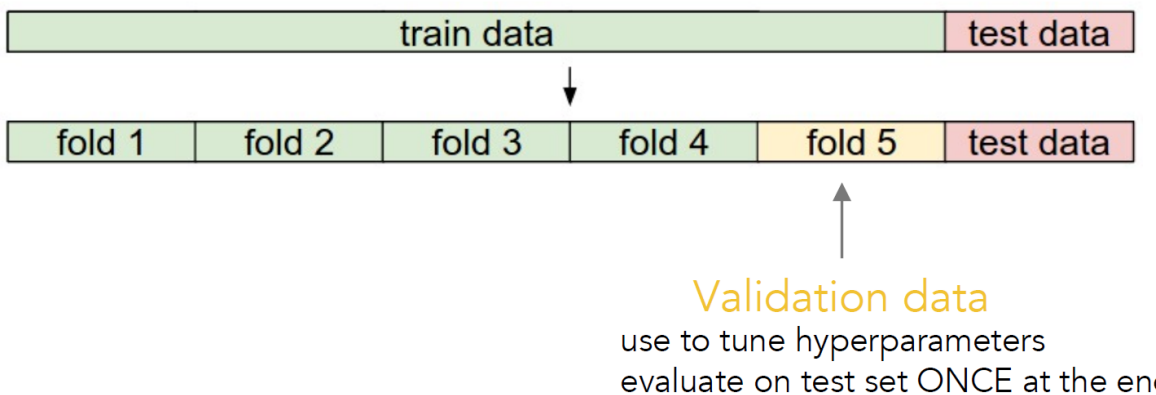


Training with validation



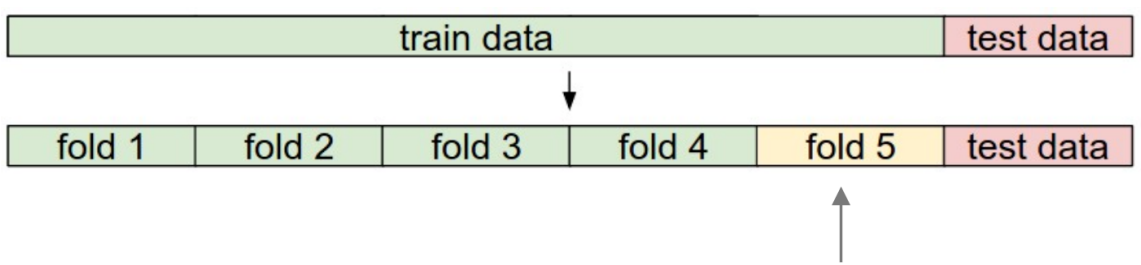
Training with validation

Validation Dataset / Cross-validation



Validation data

use to tune hyperparameters
evaluate on test set ONCE at the end



Cross-validation

cycle through the choice of which fold is the validation fold, average results.

Training problems

Some approaches to mitigating overfitting

Add regularization

Add (naturally sensible) noise to training data

Training problems

Some approaches to mitigating underfitting

“if you're not overfitting, your network isn't big enough.” – Geoffrey Hinton

Get more training data
Balance training data classes
Increase model complexity
Reduce regularization (constraints on parameter values)

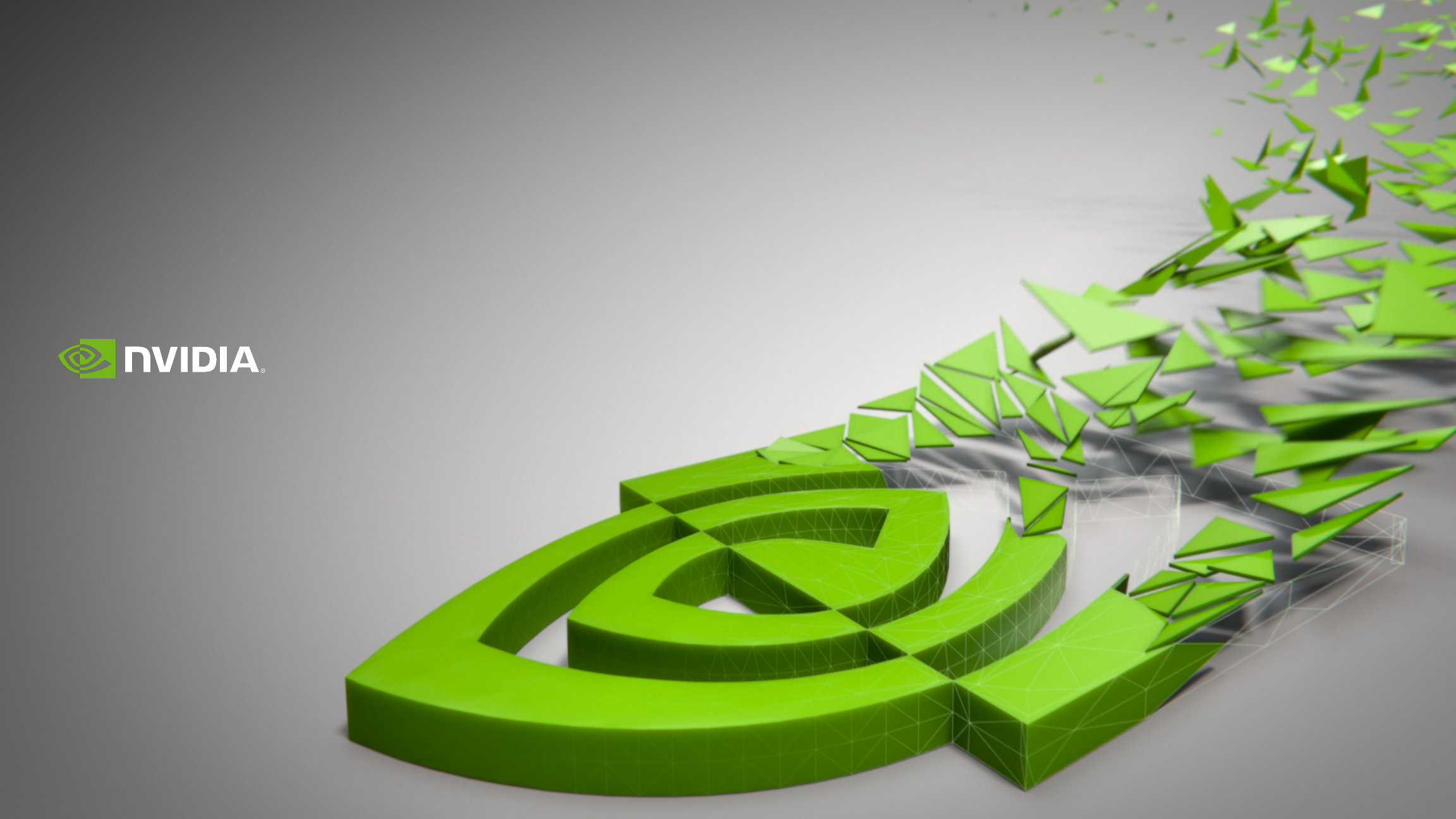
Live Demo (20 mins)

Image classification in DIGITS

Creation of training, validation and testing datasets

Definition of score and loss function

Monitoring model training


NVIDIA

Bias and Variance

- Bias: expected difference between model's prediction and truth
  - Variance: how much the model differs among training sets

- Model Scenarios
  - High Bias: Model makes inaccurate predictions on training data
  - High Variance: Model does not generalize to new datasets
  - Low Bias: Model makes accurate predictions on training data
  - Low Variance: Model generalizes to new datasets