

Figure 1: The Transformer - model architecture.

Deep Learning for NLP Workshop

Jay Urbain, PhD

jay.urbain@gmail.com

<https://www.linkedin.com/in/jayurbain/>

Deep Learning for NLP Workshop

- Deep Learning Intro
- Introduction to NLP
 - Lab 1.1 - spaCy
- Word Representations & Embeddings
 - Lab 2 - Word vectors
- Deep Learning for NLP
 - MLP, CNN, RNN/LSTM, Attention
 - Lab 3.1 PyTorch Basics
 - Lab 3.2 PyTorch Machine Learning
- Sequence learning
 - Lab 4.1 LSTMs for Sequence Learning
 - Lab 4.2 Bi-LSTMs with Attention
- Transfer Learning and Advanced Models
 - Transformer Model
 - Self-supervised pre-training, ELMo, GPT, BERT, ...
 - Lab 5 - Transfer learning: text classification ₂

WORKSHOP PERSPECTIVE

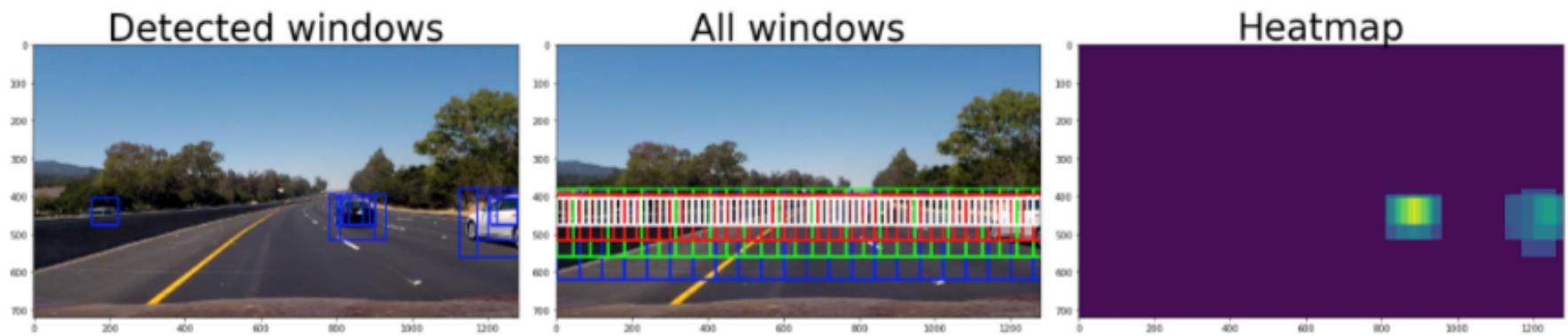
- What this workshop is:
 - An intro to Natural Language Processing (NLP) using Deep Learning
 - Lectures followed by step-by-step hands-on exercises using Jupyter Notebooks
 - Emphasis is on the hands-on exercises
 - Executable from github repository on Google Colab or local
- This workshop is not:
 - Comprehensive review of NLP
 - Deep Learning from first principles
 - Extensive programming and knowledge of APIs

EXPECTATIONS

- Assumptions:
 - Basic Python
 - Basic understanding of Machine Learning
- Nice to have:
 - Some deep learning, any NLP experience
- Consider working with a partner

HOPEFULL TAKE AWAYS

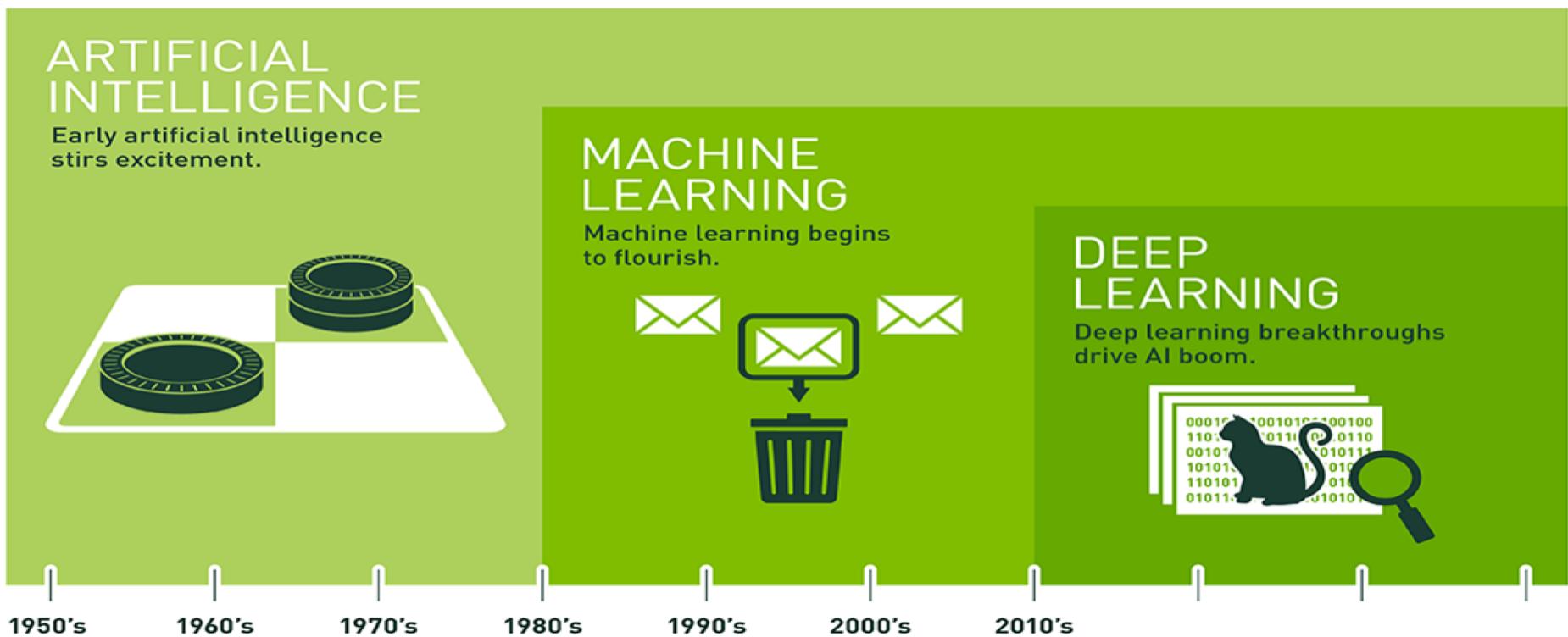
- Ability to design a deep learning workflow to conduct NLP based text tasks
- Have a starting point for creating your own NLP projects



Intro to Deep Learning

Jay Urbain, PhD
jay.urbain@gmail.com

ACCOMPLISHING COMPLEX GOALS

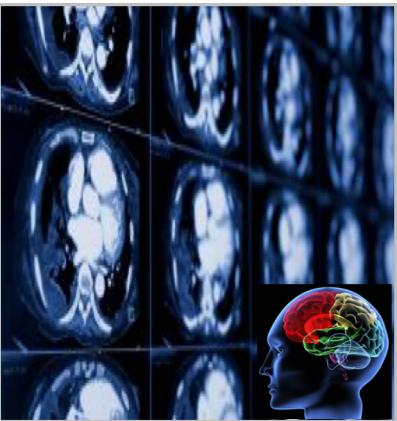


Sweeping Across Industries

Internet Services



Medicine



Media & Entertainment



Security & Defense



Autonomous Machines



- Image/Video classification
- Speech recognition
- Natural language processing

- Cancer cell detection
- Diabetic grading
- Drug discovery

- Video captioning
- Content based search
- Real time translation

- Face recognition
- Video surveillance
- Cyber security

- Pedestrian detection
- Lane tracking
- Recognize traffic signs

Deep learning = Learning representations/features

- The traditional model of pattern recognition (since the late 50's)

- Fixed/engineered features (or fixed kernel) + trainable classifier



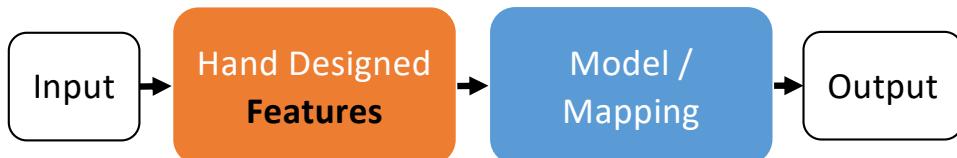
- End-to-end learning / Feature learning / Deep learning

- Trainable features (or kernel) + trainable classifier

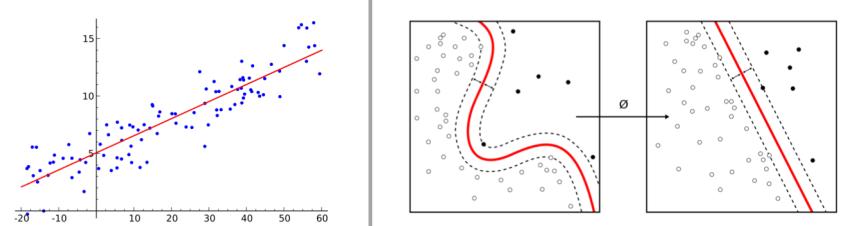


Difference in Workflow

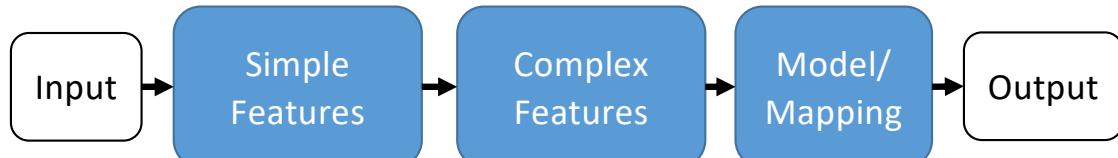
Classic Machine Learning [1990 : now]



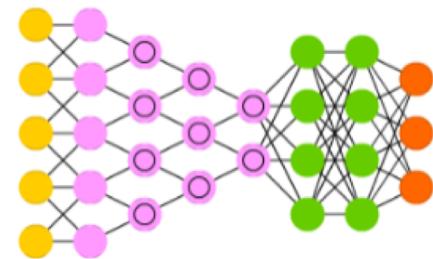
Examples [Regression and SVMs]



Deep/End-to-End Learning [2012 : now]



Example [Conv Net]

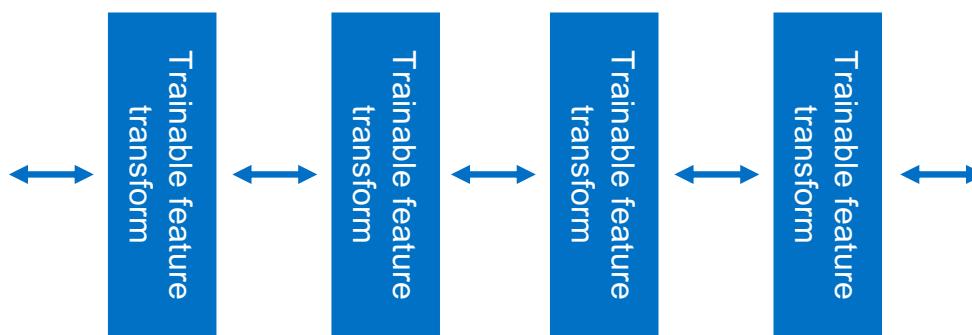


Machine learning workflow shifts from engineering features for “shallow” models to architecting deep learning models with the ability to learn hierarchical representations of features

Trainable feature hierarchy

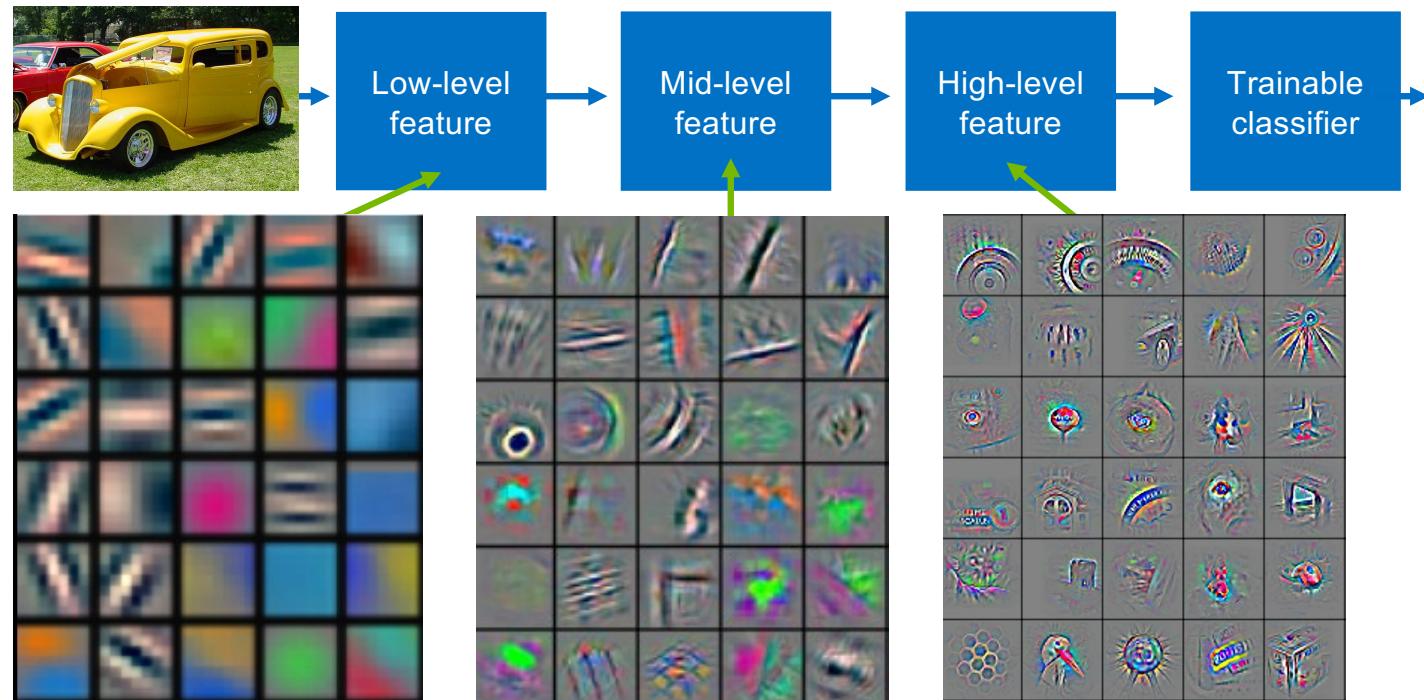
Hierarchy of representations with increasing level of abstraction. Each stage is a kind of trainable feature transform

- Image recognition
 - Pixel → edge → motif → part → object
- Text
 - Character → word → word group → clause → sentence → story/semantic understanding
- Speech
 - Sample → spectral band → sound → ... → phone → phoneme → word



Deep learning = learning hierarchical representations

It's **deep** if it has **more than one stage** of non-linear feature transformation

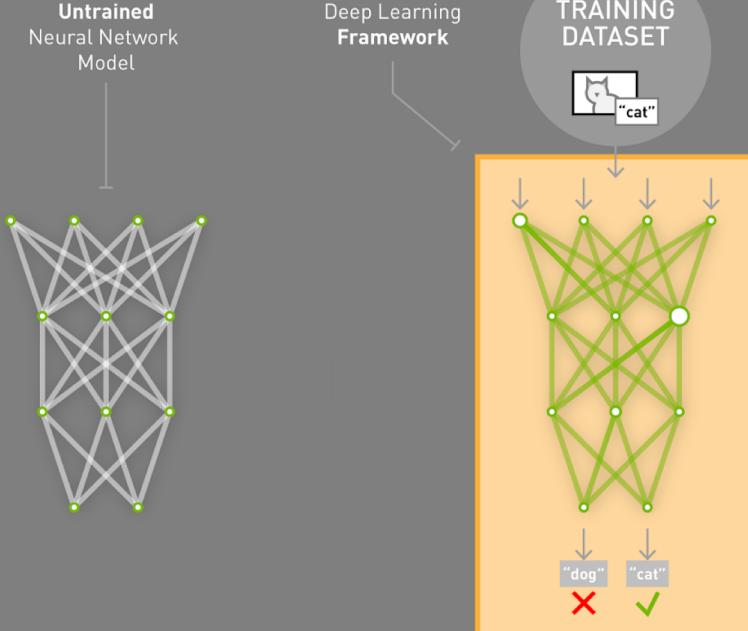


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

DEEP LEARNING

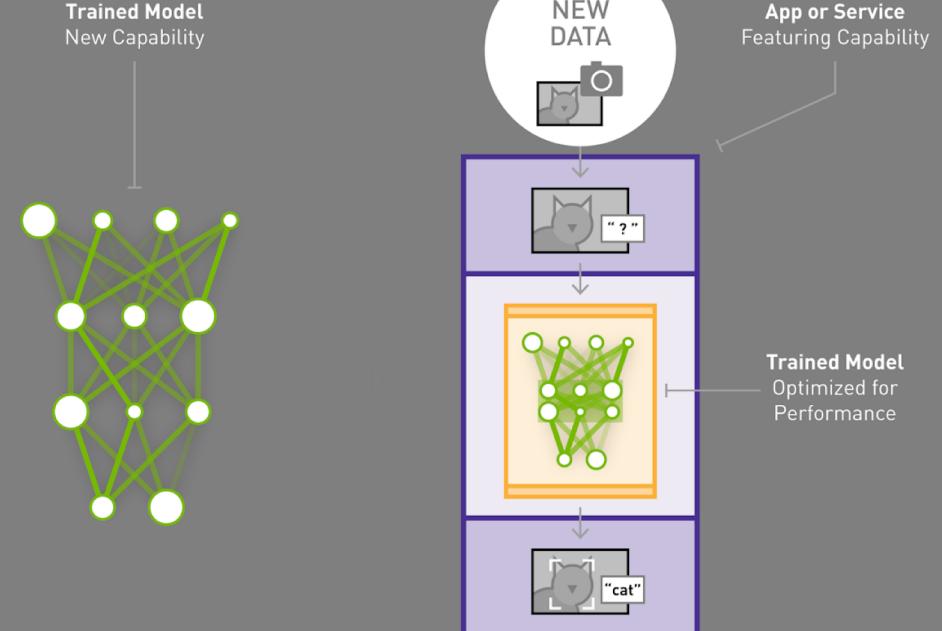
TRAINING

Learning a new capability
from existing data



INFERENCE

Applying this capability
to new data



Rationale for Deep Learning

Costs of acquiring and storing large quantities of heterogeneous data have dropped significantly.

The ability to extract actionable knowledge from such datasets has become critical for companies to maintain a competitive advantage.

Significant improvements in deep learning algorithms enabled by GPU processing provide new application opportunities across industries.

Deep Learning has set new performance standards in many machine learning applications.