Introduction to AI

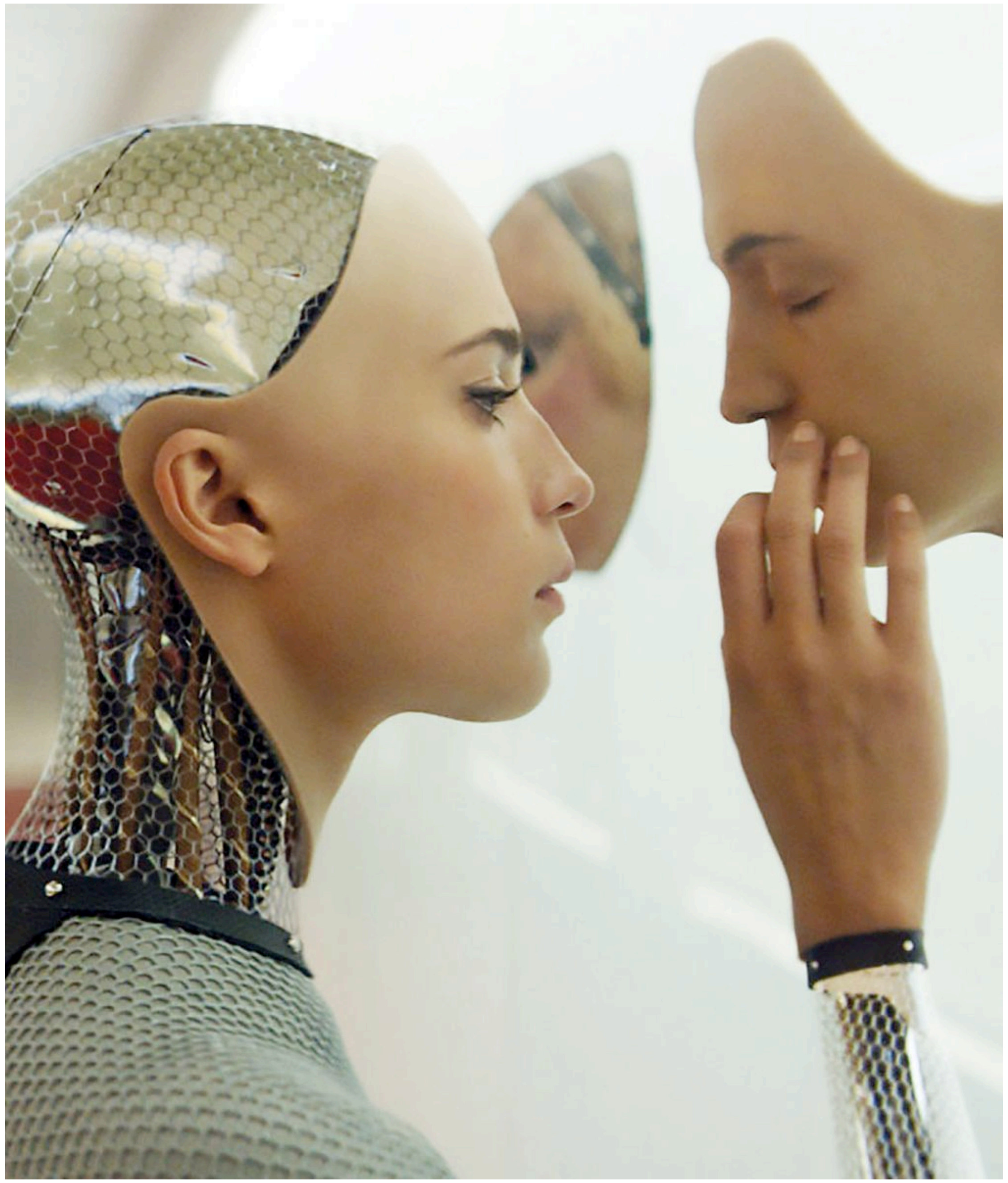
Math in Machine Learning seminar (MiML)
McGill Math and Stats (McMaS)

Background AI

- Artificial Intelligence is loosely defined as intelligence exhibited by machines
- Operationally: R&D in CS academic sub-disciplines: Computer Vision, Natural Language Processing (NLP), Robotics, etc

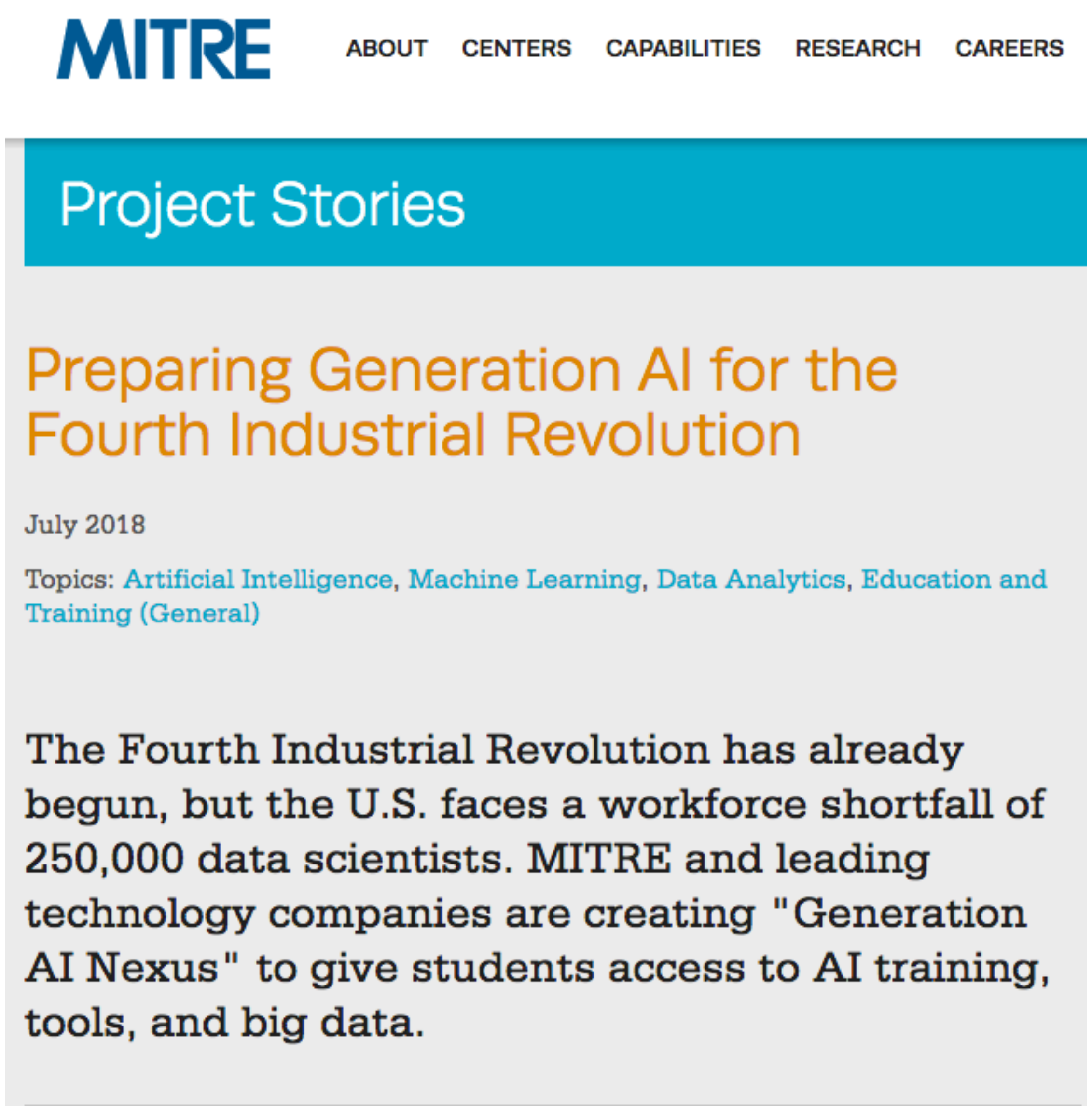
Artificial General Intelligence (AGI)

- AI : specific tasks,
- AGI : general cognitive abilities.
- (AGI) is a small research area within AI: build machines that can successfully perform *any* task that a human might do
- On account of this ambitious goal, AGI has high visibility, disproportionate to its size or present level of success, among futurists, science fiction writers, and the public.



Perspectives on Research in Artificial Intelligence and Artificial General Intelligence Relevant to DoD

taken from a study by JASON for the Department of Defence (Dod)



Project Stories

Preparing Generation AI for the Fourth Industrial Revolution

July 2018

Topics: Artificial Intelligence, Machine Learning, Data Analytics, Education and Training (General)

The Fourth Industrial Revolution has already begun, but the U.S. faces a workforce shortfall of 250,000 data scientists. MITRE and leading technology companies are creating "Generation AI Nexus" to give students access to AI training, tools, and big data.

Historical Context

- “AI” coined in 1956.
- Perceptrons 1960 implied machines could learn from data
- Decline in 1969 - perceptron no universal function approximator - only linear discriminator, can’t learn XOR
- 1980 resurgence in AI “expert systems”. Learning rules. Petered out
- 1990s academic AI in doldrums
- Improved computers led, in 1997 to IBM Deep Blue beats champion Gary Kasparov in chess.   Chess, *once believed to require human intelligence, fell to a special-purpose very fast search algorithm.*

Don’t confuse AI with AGI

- 1997 NYT in response to Deep Blue: “to play a decent game of Go [requires human intelligence]” when that happens, will be a sign that AI is “as good as the real thing”

- 2016 NYT wrong: Google’s AlphaGO beats world Champion Lee Sedol. Did not involve breakthrough - also using hybrid of DNN with massively parallel tree-search and Reinforcement Learning

- DNN require massive amounts of data which can be found

    - labelled on the internet,

    - or in the databases of private companies, like Facebook or Google,

    - generated from a fast computer playing a lifetime of games

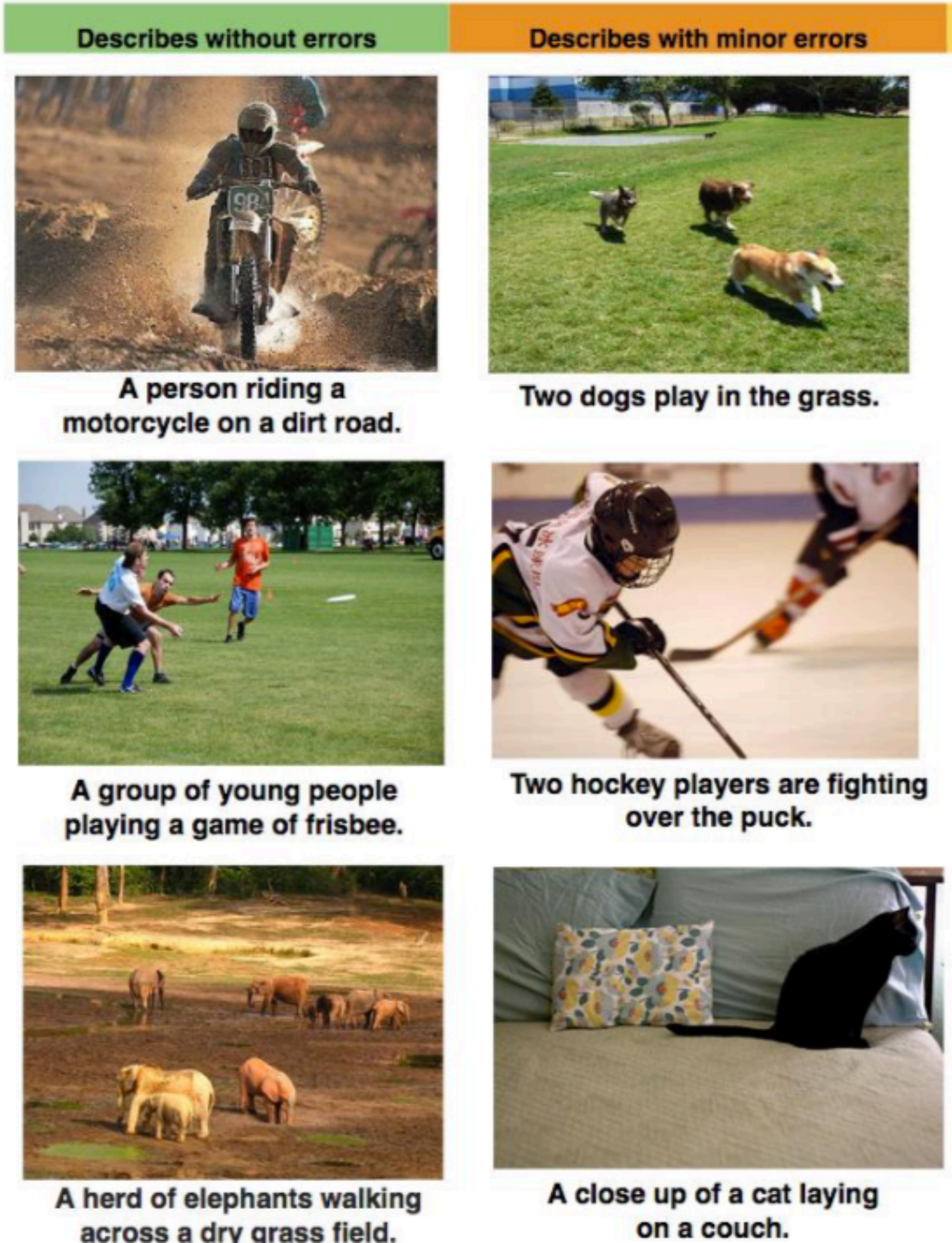
2010: Deep Learning Revolution

- Neural Networks have been around for half a century. Popular in the 1990’s for solving simple tasks.
- Starting around 2010, new hardware, Graphics Processor Units (GPU)s, became available, which allowed for much larger, and deeper networks.
- large labelled data sets become available, allowed for training.

2010: Deep Learning Revolution

- The large data set ImageNet was available in 2005.
- In 2012 Alexnet, trained on GPUs, won the 2012 ImageNet competition, with an error of 15.3%, more that 10% better than the runner up. Canadian (U Toronto) team: Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever.
- Between 2011 and 2015, error rate for image captioning by computer fell from 25% to 3%, better than accepted human figure of 5%

more than 95% prediction correct caption (green column)



Describes without errors

Describes with minor errors

A person riding a motorcycle on a dirt road.

Two dogs play in the grass.

A group of young people playing a game of frisbee.

Two hockey players are fighting over the puck.

A herd of elephants walking across a dry grass field.

A close up of a cat laying on a couch.

DNN and Images

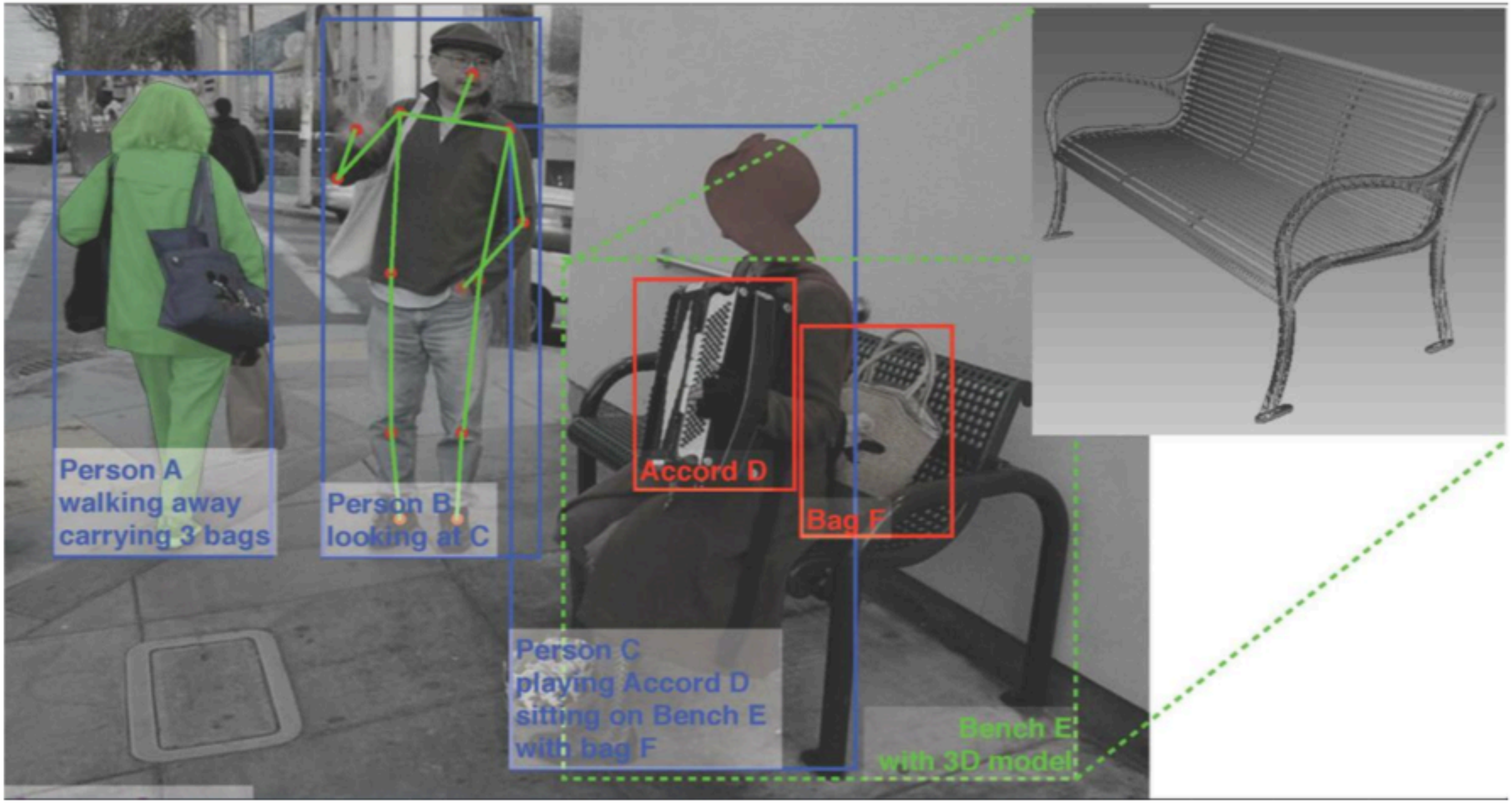


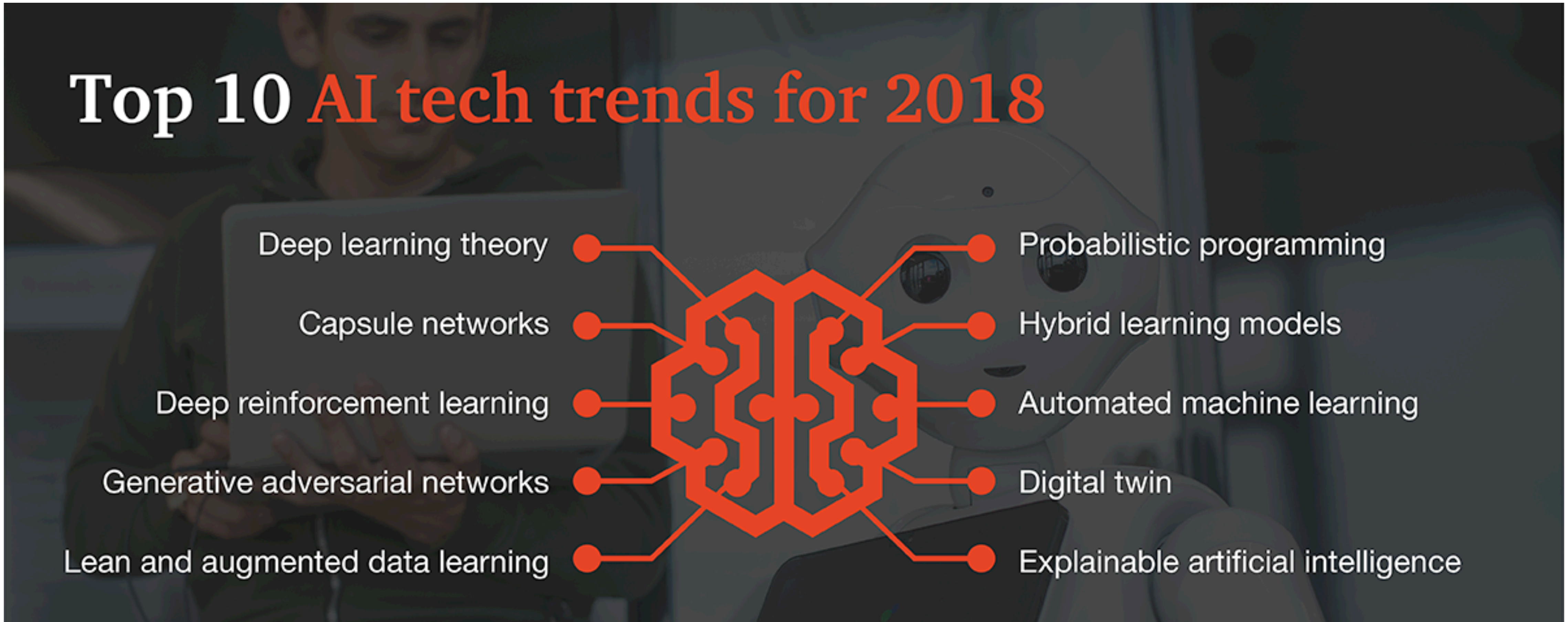
Figure 2. Beyond captioning and object identification, computer vision using DNNs is capable of (a) image segmentation—identifying the pixels that constitute Person A, (b) pose estimation—using a knowledge of human anatomy to construct a likely three-dimensional pose of Person B, (c) associating groups of objects, as that Person C is playing an accordion, (d) recognizing and constructing 3-D models of partially hidden objects, as here the bench. Source: Jitendra Malik.

DNN exceed human performance in:

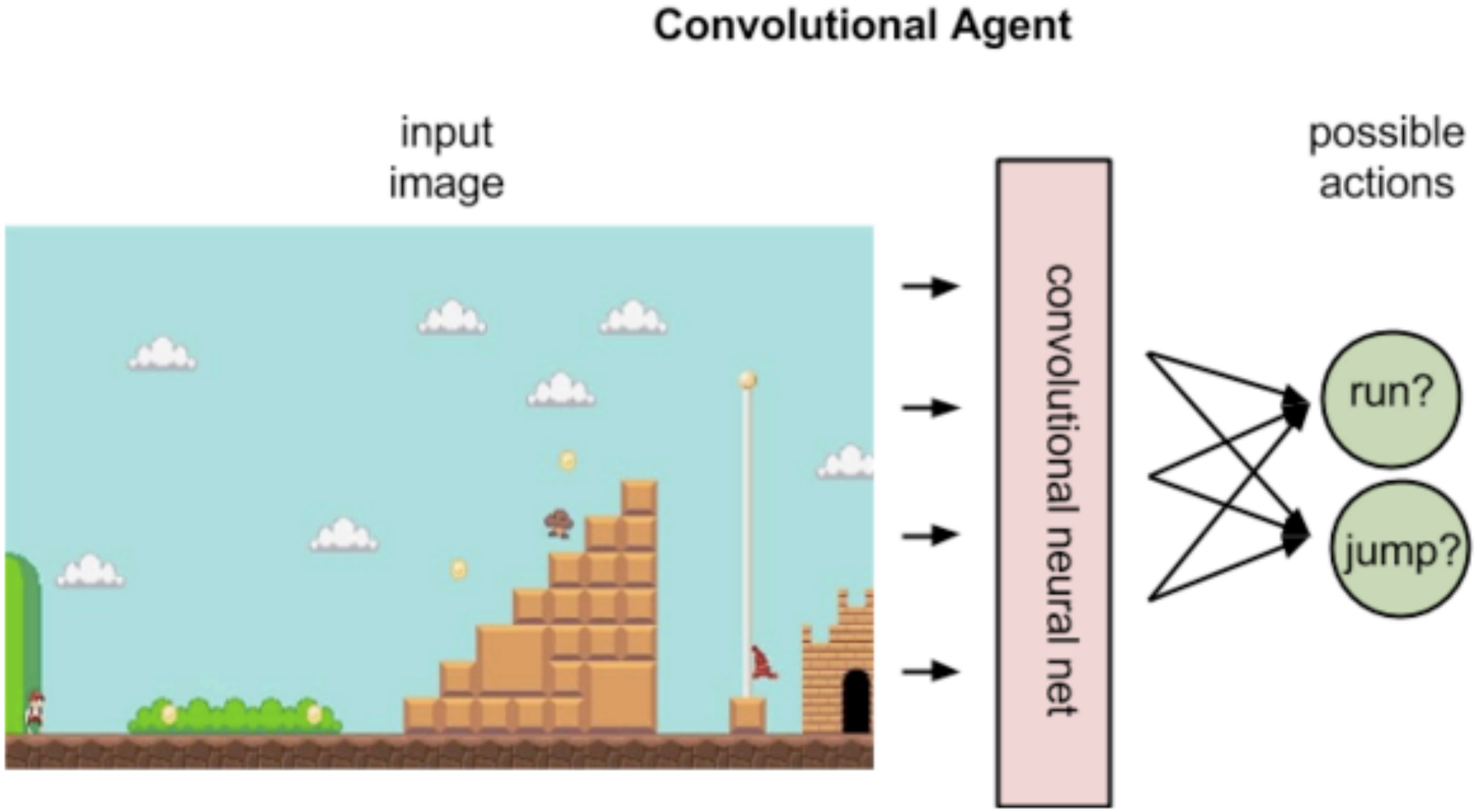
- some kinds of image recognition
- spoken word recognition
- the game of Go (long thought to require generalized human intelligence—AGI)
- self-driving cars: now more limited by policy than tech.

rapidly advancing areas

- Reinforcement Learning
- graphical and Bayes models, esp. with probability programming models
- generative models (creating artificial images)
- more likely DL will become essential building block of a hybrid approach

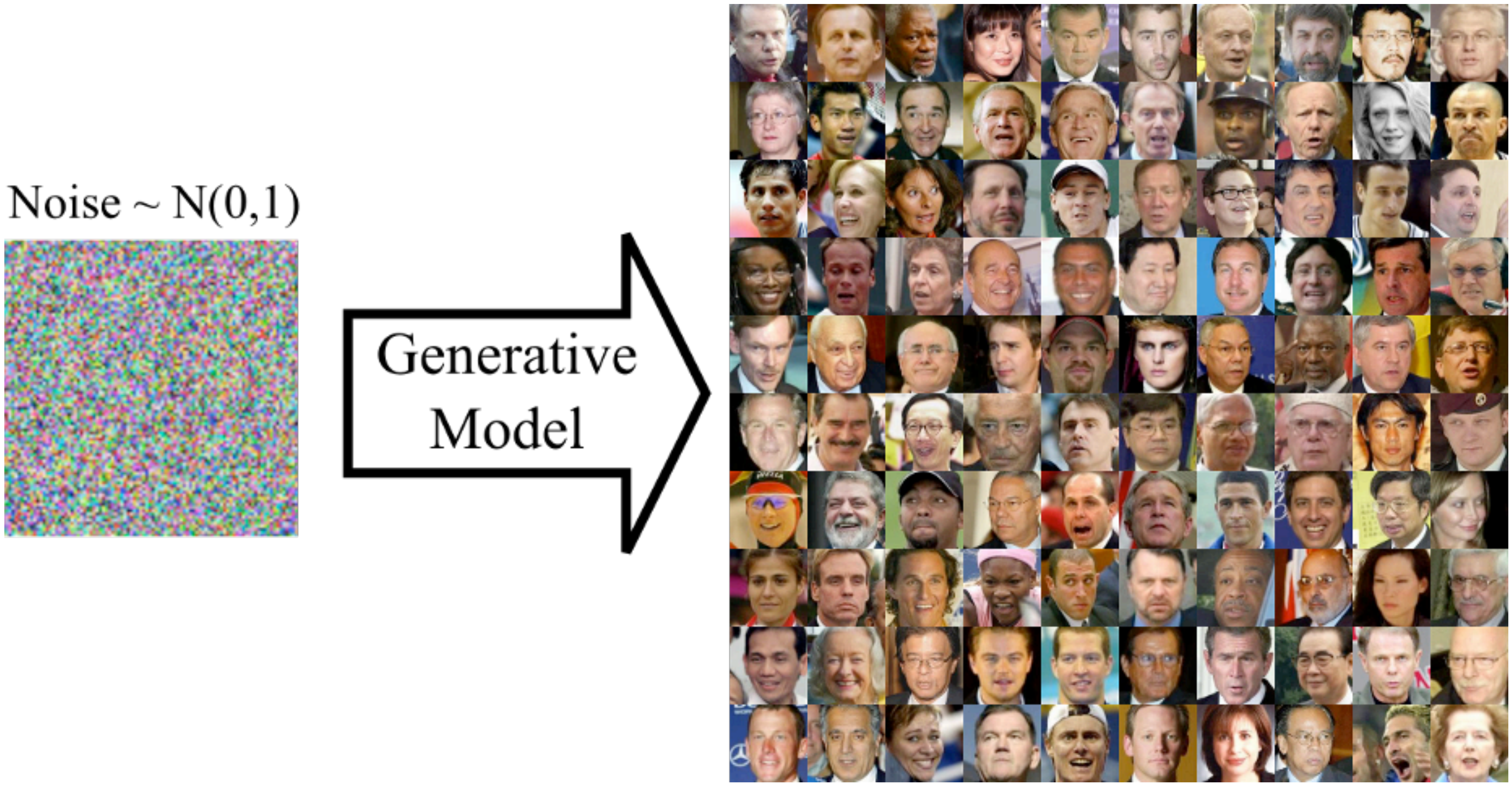

Top 10 AI tech trends for 2018
Deep learning theory
Capsule networks
Deep reinforcement learning
Generative adversarial networks
Lean and augmented data learning
Probabilistic programming
Hybrid learning models
Automated machine learning
Digital twin
Explainable artificial intelligence

Reinforcement Learning

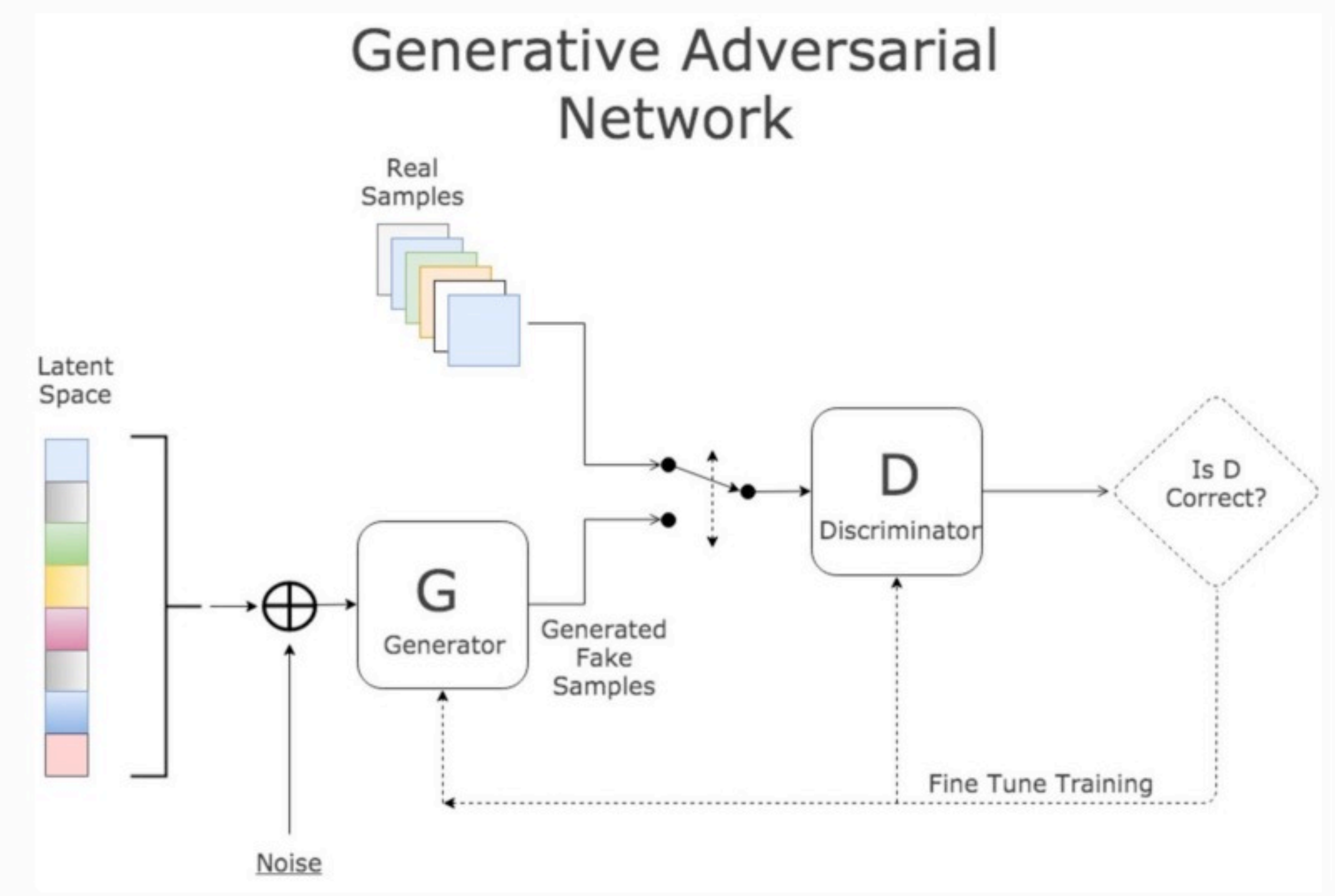


- Learn how to play Atari from raw image pixels.
- Learn how to beat Go champions (Huge branching factor)
- Robots learning how to walk
- Big in Montreal: Google DeepMind and Microsoft Research both work in this area

Generative Models



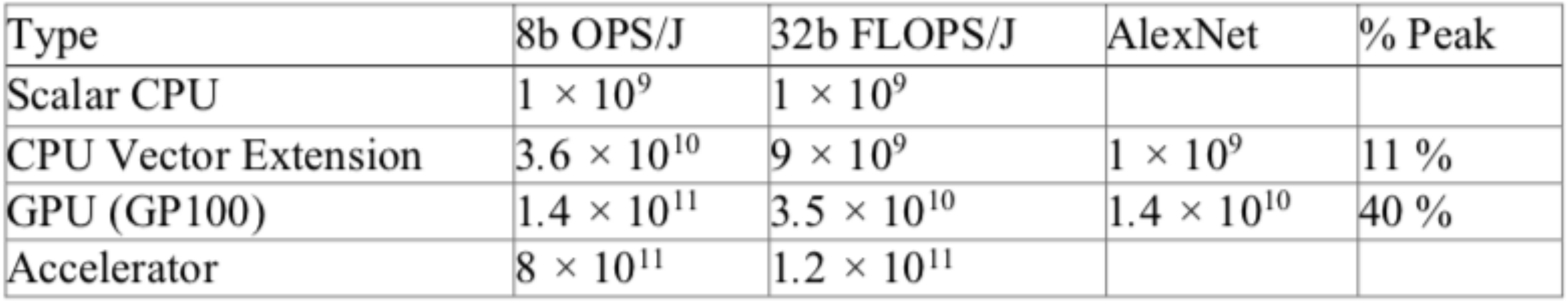
A generative model takes a input random vectors and outputs realistic images (of a certain class)



- discriminator: has learned what a picture looks like.
- generator: tries to generate a believable picture.

Hardware

- Both training (finding the best weights) and inference (evaluating the output of the network on a data point) are computationally intensive. Effort is measured in Joules.
- Hardware, software, and algorithms have evolved together.
- Currently, need to use Graphics Processing Units (GPU)s, rather than CPUs.

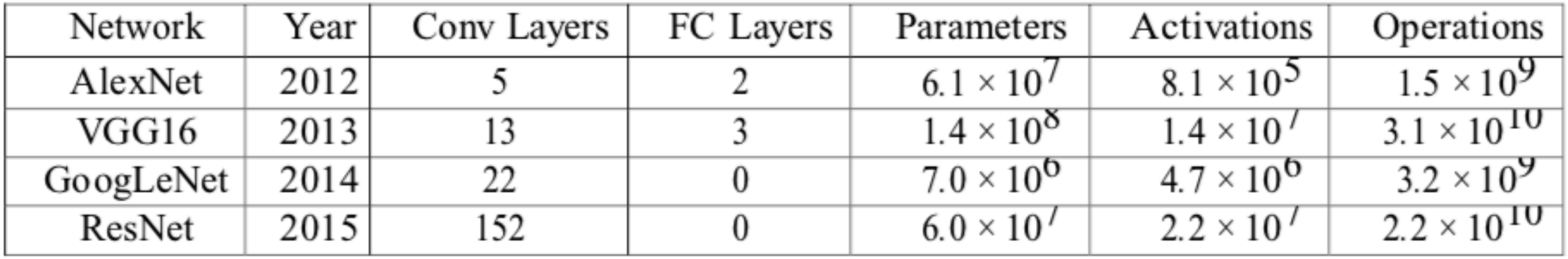


Type	8b OPS/J	32b FLOPS/J	AlexNet	% Peak
Scalar CPU	1×10^9	1×10^9		
CPU Vector Extension	3.6×10^{10}	9×10^9	1×10^9	11 %
GPU (GP100)	1.4×10^{11}	3.5×10^{10}	1.4×10^{10}	40 %
Accelerator	8×10^{11}	1.2×10^{11}		

Hardware

- Want to do inference on mobile devices,
- need custom architectures, and custom hardware.
- Currently engineering practice to take trained networks and make them smaller. Also build custom chips with power source
- Research problem: design and train architectures with efficient inference in mind

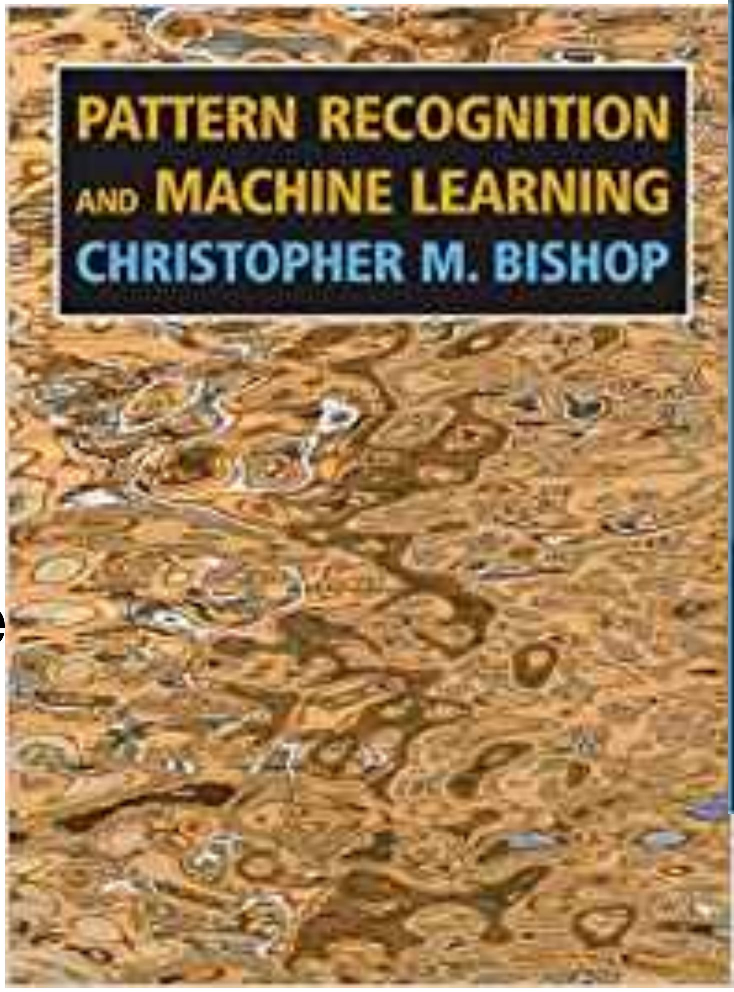
Table 1. Evolution of DNNs for ImageNet.

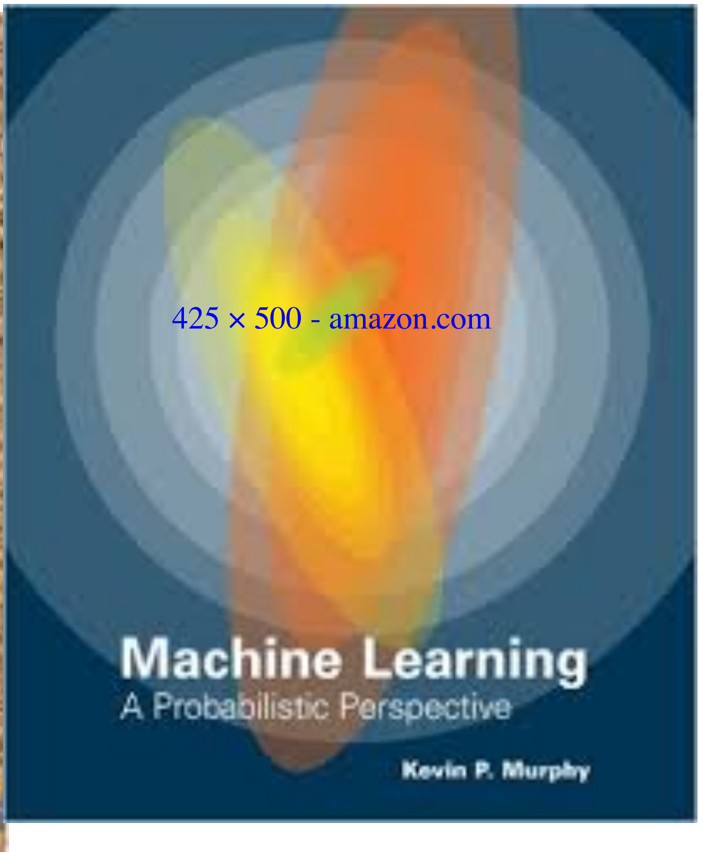


Network	Year	Conv Layers	FC Layers	Parameters	Activations	Operations
AlexNet	2012	5	2	6.1×10^7	8.1×10^5	1.5×10^9
VGG16	2013	13	3	1.4×10^8	1.4×10^7	3.1×10^{10}
GoogLeNet	2014	22	0	7.0×10^6	4.7×10^6	3.2×10^9
ResNet	2015	152	0	6.0×10^7	2.2×10^7	2.2×10^{10}

Machine Learning vs DL

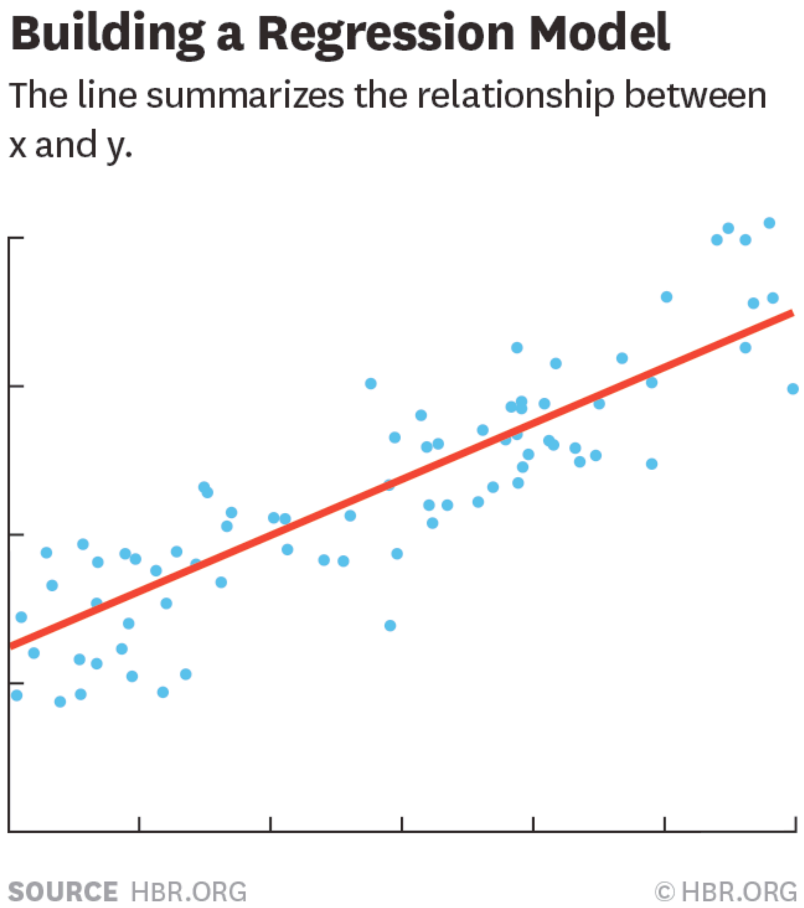
- Traditional Machine Learning (ML) can’t compete with the raw performance of Deep Learning
- However ML has performance guarantees which are important in the many applications where errors are costly.





Error Estimates

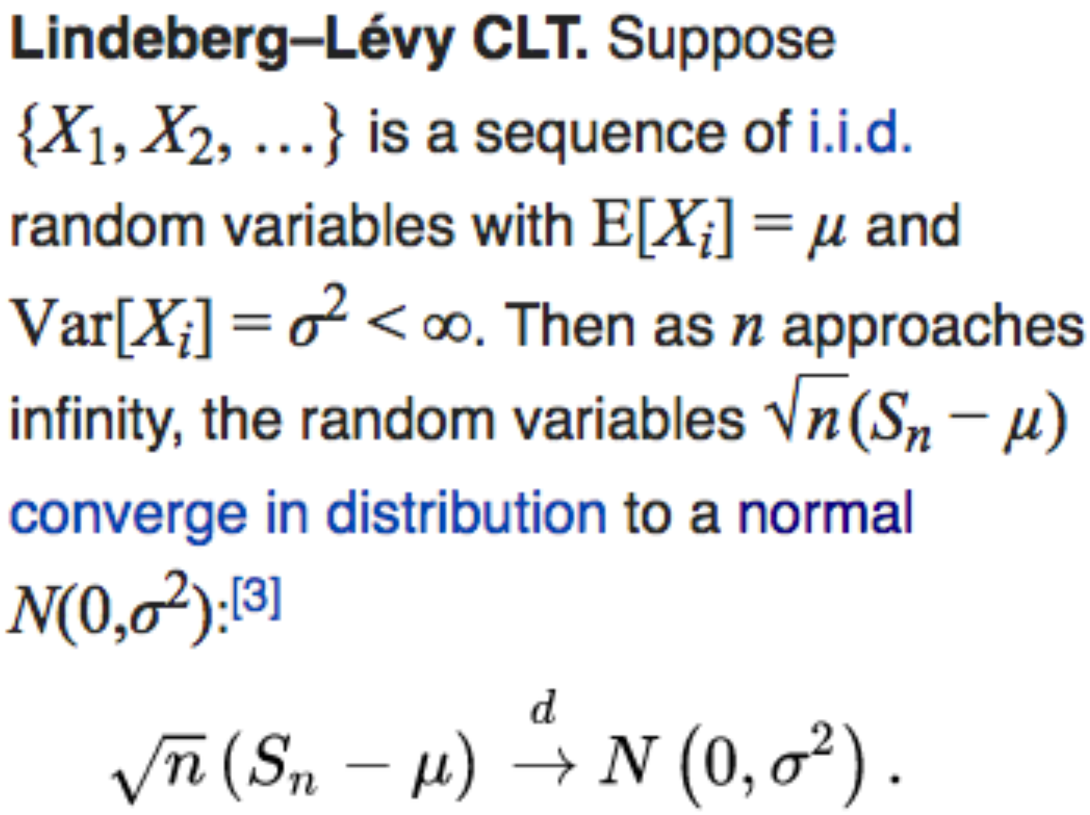
- Using probability (Central Limit Theorem) and linear or parametric models, can fit data, and also estimate the probability of an error
- Deep Learning models lack these estimates on errors!



Building a Regression Model

The line summarizes the relationship between x and y.

SOURCE HBR.ORG © HBR.ORG



Lindeberg–Lévy CLT. Suppose $\{X_1, X_2, \ldots\}$ is a sequence of i.i.d. random variables with $\mathrm{E}[X_i] = \mu$ and $\mathrm{Var}[X_i] = \sigma^2 < \infty$. Then as n approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to a normal $N(0,\sigma^2)$:[3]

$$\sqrt{n}\left(S_n - \mu\right) \xrightarrow{d} N\left(0, \sigma^2\right).$$

Neural Network Architecture

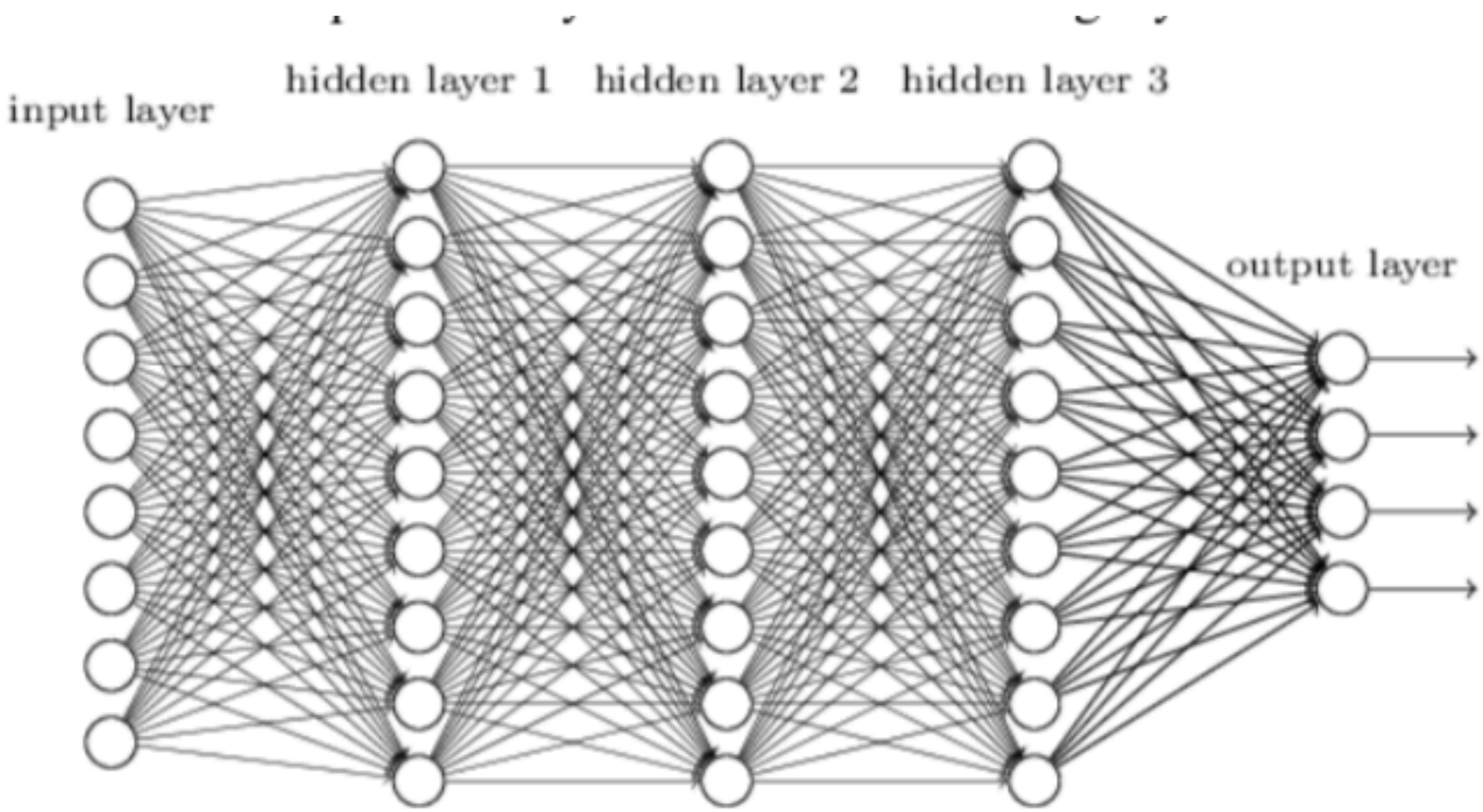


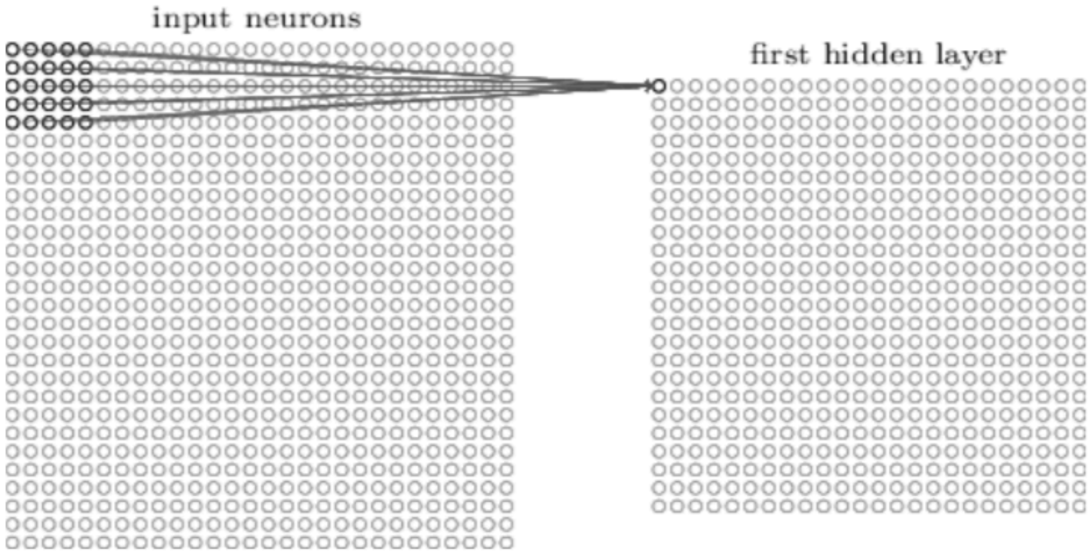
Figure 3. A simple deep neural network (DNN). This network consists of five layers: an input layer with eight neurons, three hidden layers with nine neurons each, and an output layer with four neurons. Layers in modern neural networks have thousands to millions of neurons in dozens to hundreds of layers. Source: neuralnetworksanddeeplearning.com/chapt6.html.

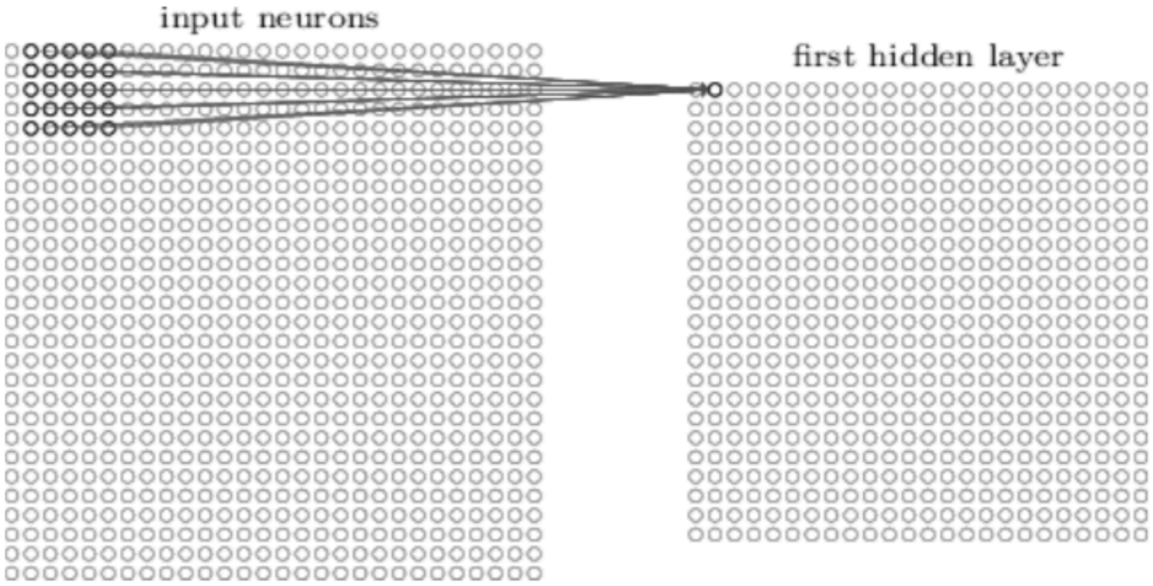
$$y = \sum w_i x_i + b_i$$

$$z = \sigma^{ReLU}(y) = \max(y, 0)$$

Convolutional Neural Nets

- Deep NN: allows different weights everywhere.
- Convolutional NN: special case, for images, where weights are nonzero only for nearby neighbors (at the input level and later). In addition, for each layer, the pattern of the weights is the same at every location.
- Significantly reduces the total number of weights per layer, allowing for much deeper networks.





More architecture

In summary, Figure 9 shows a schematic of what might be a typical convolutional layer in a network. The input is a 28x28 pixel image. The image is processed with a 5x5 convolutional kernel and a stride length of one pixel, yielding 24x24 neuron feature maps. Here there are are three feature maps, each with its unique set of weights. Each feature map is pooled with a 2x2 pooling layer, yielding three 12x12 feature maps.

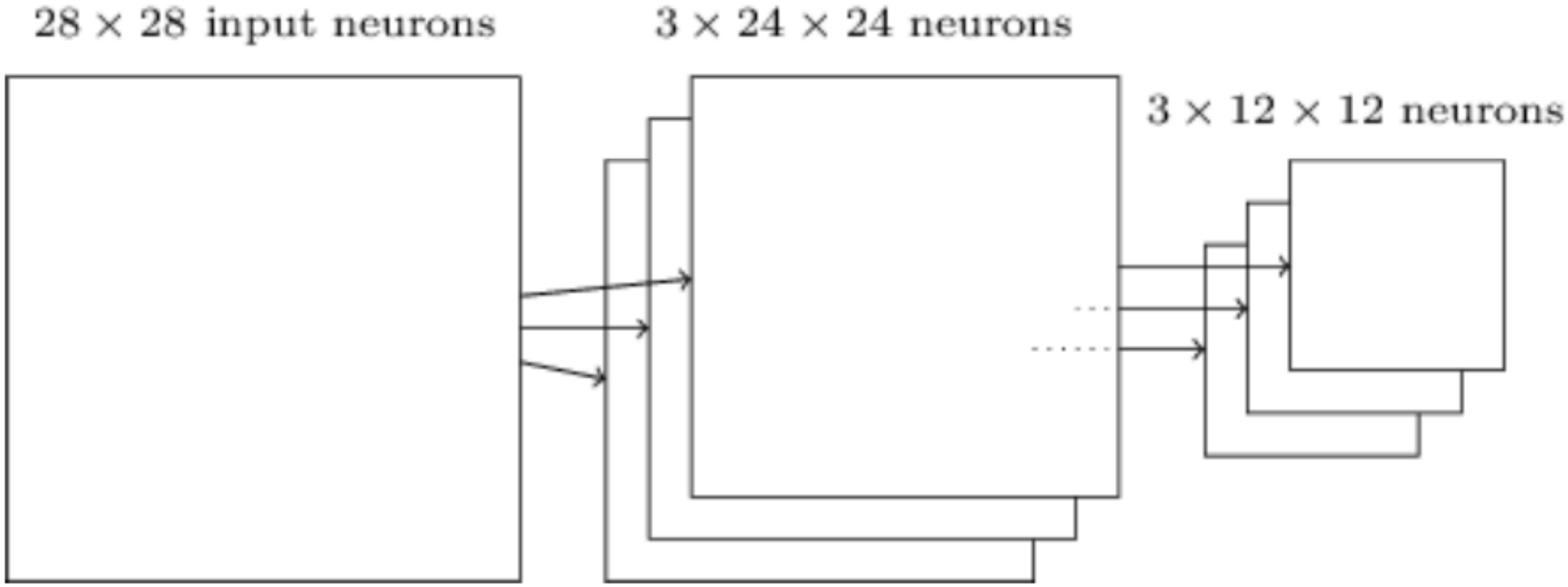


Figure 9. A schematic of a typical layer in a convolutional network. The input is a 28x28 pixel image. The image is processed with a 5x5 convolutional kernel and a stride length of one pixel, yielding 24x24 neuron feature maps. Here there are are three feature maps, each with its unique set of weights. Each feature map is pooled with a 2x2 pooling layer, yielding three 12x12 feature maps. Source: neuralnetworksanddeeplearning.com/chapt6.html .

Training

Given data x_i, labels $y(x_i)$ and network $u(x;w)$ with weights w

$$\min_w L(w) \equiv \frac{1}{n} \sum_i \ell(u(x_i;w), y(x_i))$$

- sum is over a large number (millions) of data points. Instead approximate sum by a random subset (mini-batch) of hundreds of data points.

- minimize over weights by stochastic gradient descent (SGD): taking a small step in the gradient direction. Step size is called *learning rate.*

- SGD has a faster version: Nesterov's Momentum, which adds a momentum term to the update.

- The gradient is computed (automatically by software) using the chain rule

$$w^{n+1} = w^n + dt_n \nabla_w L(w)$$

Technical Details

- Training CNNs is buggy. Gradients can be zero, causing training to stall, or can blow up.

- Lots of hacks or heuristics used to help.

- Regularization: the loss function is non-convex. So add an extra term to make it convex, and keep the weights small. Called weight decay.

- Data augmentation: cutout: change the images at each stage, keeping same labels

- Dropout: randomly set half the weights to zero at each iteration. (Heuristic which helps)

- batch normalization: normalize the input data to each neuron, to be mean zero var = 1.

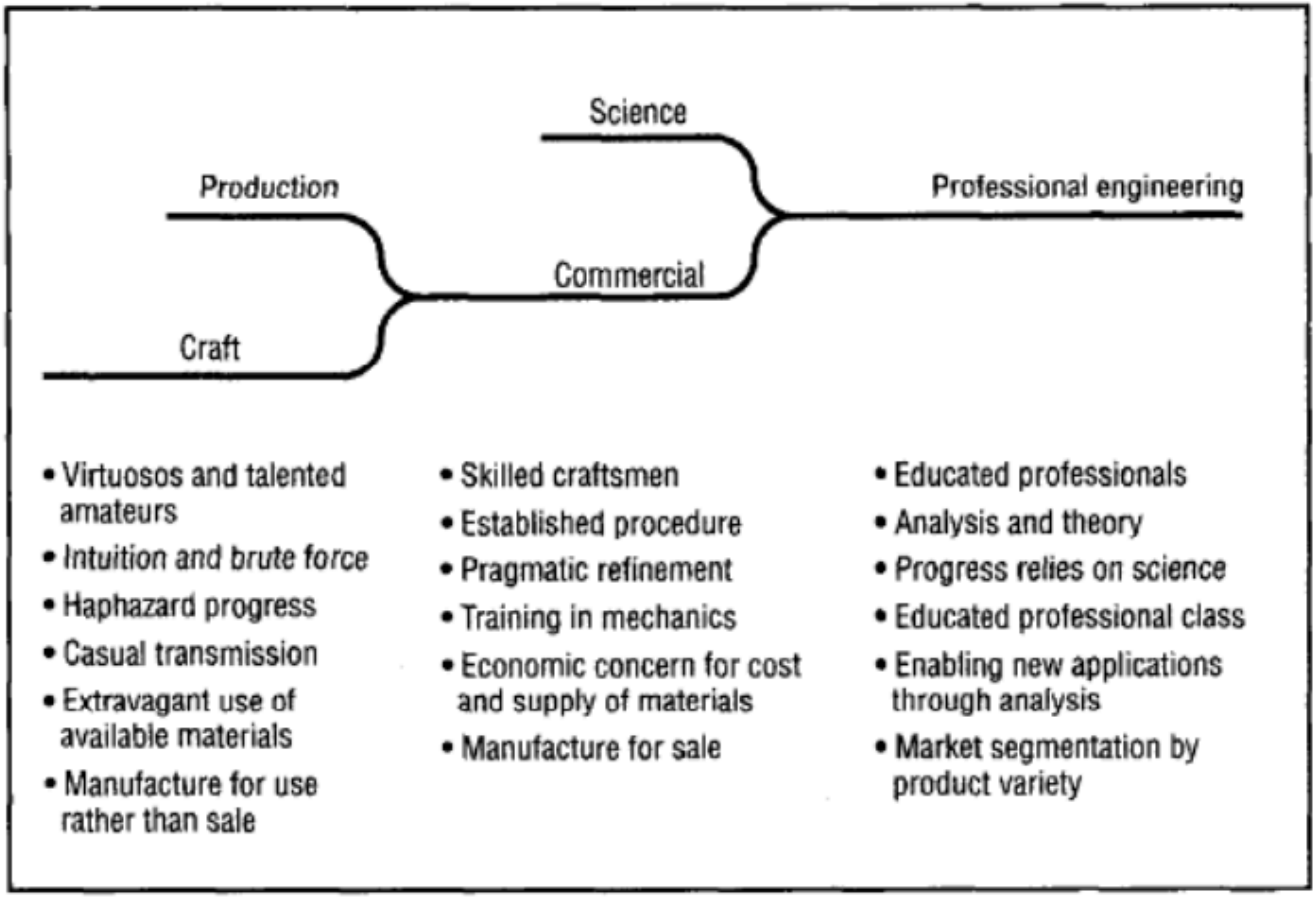
$$J(w) = L(w) + \epsilon|w|_2^2$$

$$J(w) = L(w) + \epsilon|w|_1$$

Challenges for rigorous deep learning

“it is not clear that the existing AI paradigm is immediately amenable to any sort of software engineering validation and verification. This is a serious issue, and is a potential roadblock to DoD’s use of these modern AI systems, especially when considering the liability and accountability of using AI in *lethal systems*.” JASON report (italics mine)

Evolution of engineering discipline



- Virtuosos and talented amateurs
- Intuition and brute force
- Haphazard progress
- Casual transmission
- Extravagant use of available materials
- Manufacture for use rather than sale

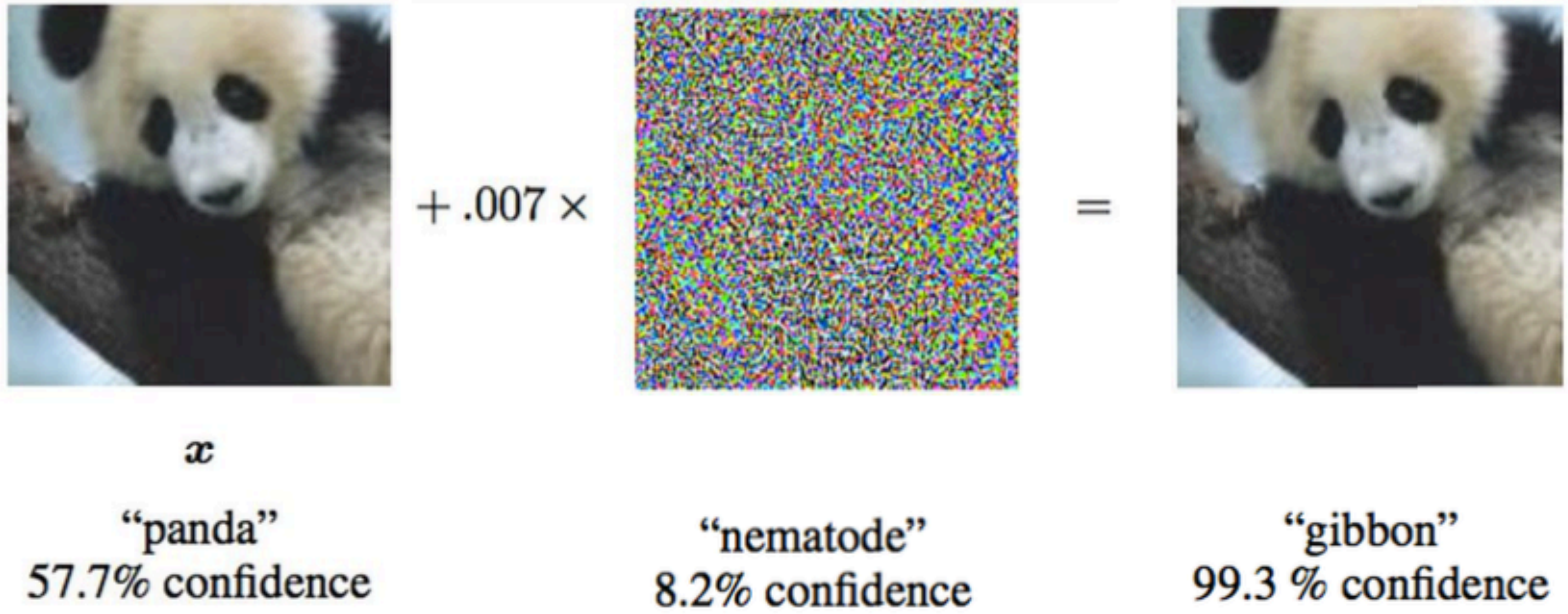
- Skilled craftsmen
- Established procedure
- Pragmatic refinement
- Training in mechanics
- Economic concern for cost and supply of materials
- Manufacture for sale

- Educated professionals
- Analysis and theory
- Progress relies on science
- Educated professional class
- Enabling new applications through analysis
- Market segmentation by product variety

Importance of “-ilities”

- Reliability
- maintainability
- accountability
- verifiability
- evolvability
- attackability

Challenge: Adversarial Examples



gradient vector from a particular panda to the nearest gibbon boundary

Goodfellow, Explaining and Harnessing Adversarial Examples, 2015

Hot current topic: next time we will talk about our progress on it.