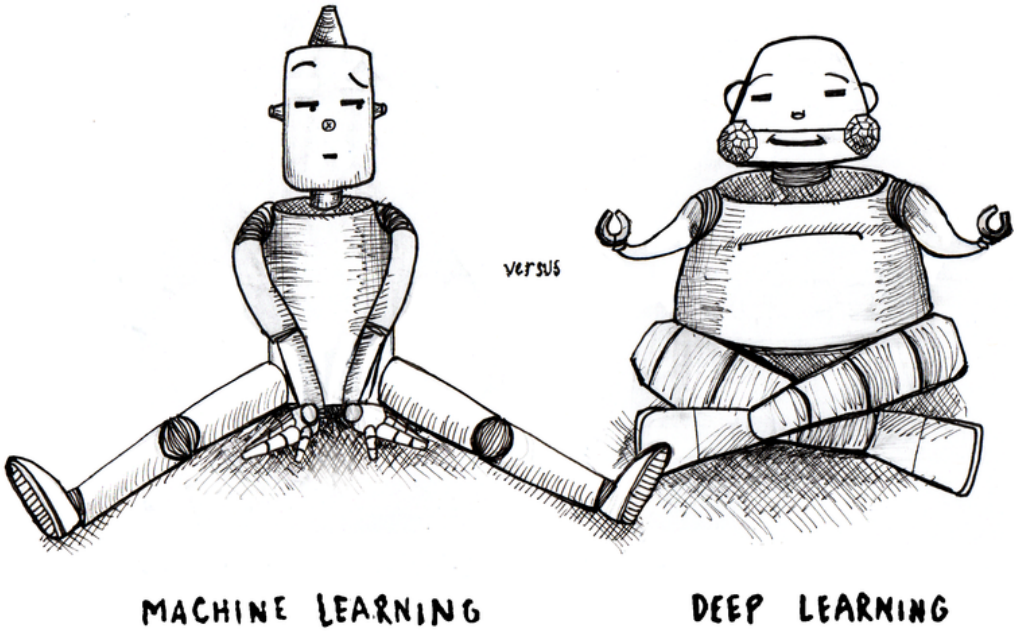
UNIVERSITY OF TRENTO - Italy

FONDAZIONE BRUNO KESSLER

From Machine Learning to Deep Learning

Elisa Ricci

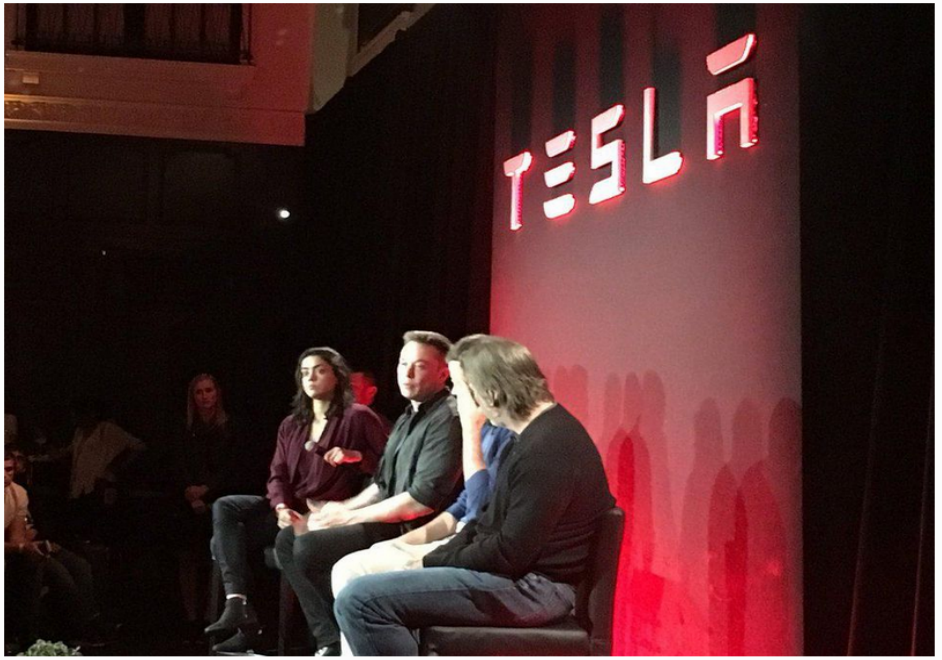
University of Trento & FBK



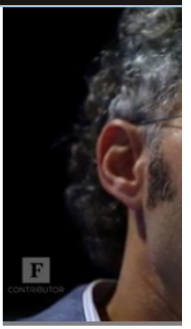
NIPS

The New Burning Man – The AI Conference That Sold Out In 12 Minutes

William Falcon Contributor ⓘ
Science
I write about AI, its business use and ethical use.



Elon Musk at Tesla private party hosted at NIPS WILLIAM FALCON



The AI Revolution

FORTUNE SUBSCRIBE

Illustration by Justin Metz

SEPTEMBER 28, 2016, 5:00 PM EDT

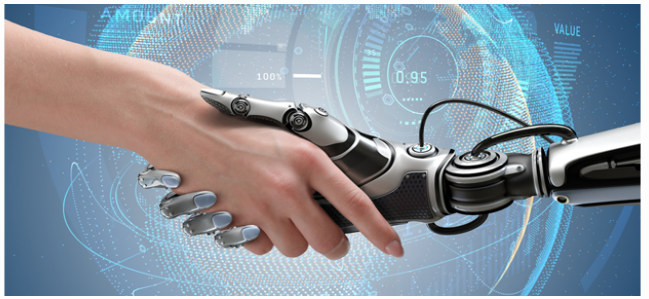
WHY DEEP LEARNING IS SUDDENLY CHANGING YOUR LIFE

Decades-old discoveries are now
and will soon transfo

Over the past four years, readers have doubt
a wide range of everyday technologies.

Most obviously, the speech-recognition func

Share Share Tweet G+1



A survival guide for the coming AI revolution

By Natalie Rens, Juxi Leitner | Mar 03, 2017

This article first appeared on The Conversation.

If the popular media isto be believed, artificial intelligence is coming to steal your job and threaten life as we know it. If we do not prepare now, we may face a future where AI runs free and dominates humans in society.

The AI revolution is indeed underway. To ensure you are prepared to make it through the times ahead, we've created a handy survival guide for you.

Step 1: Recognizing AI

The first step in every conflict is knowing your target. It is crucial to acknowledge that AI is not in the future; it is already here.

MOST POPULAR ARTICLES

Driving the future of smart citie

When digital service teams hit hurdles

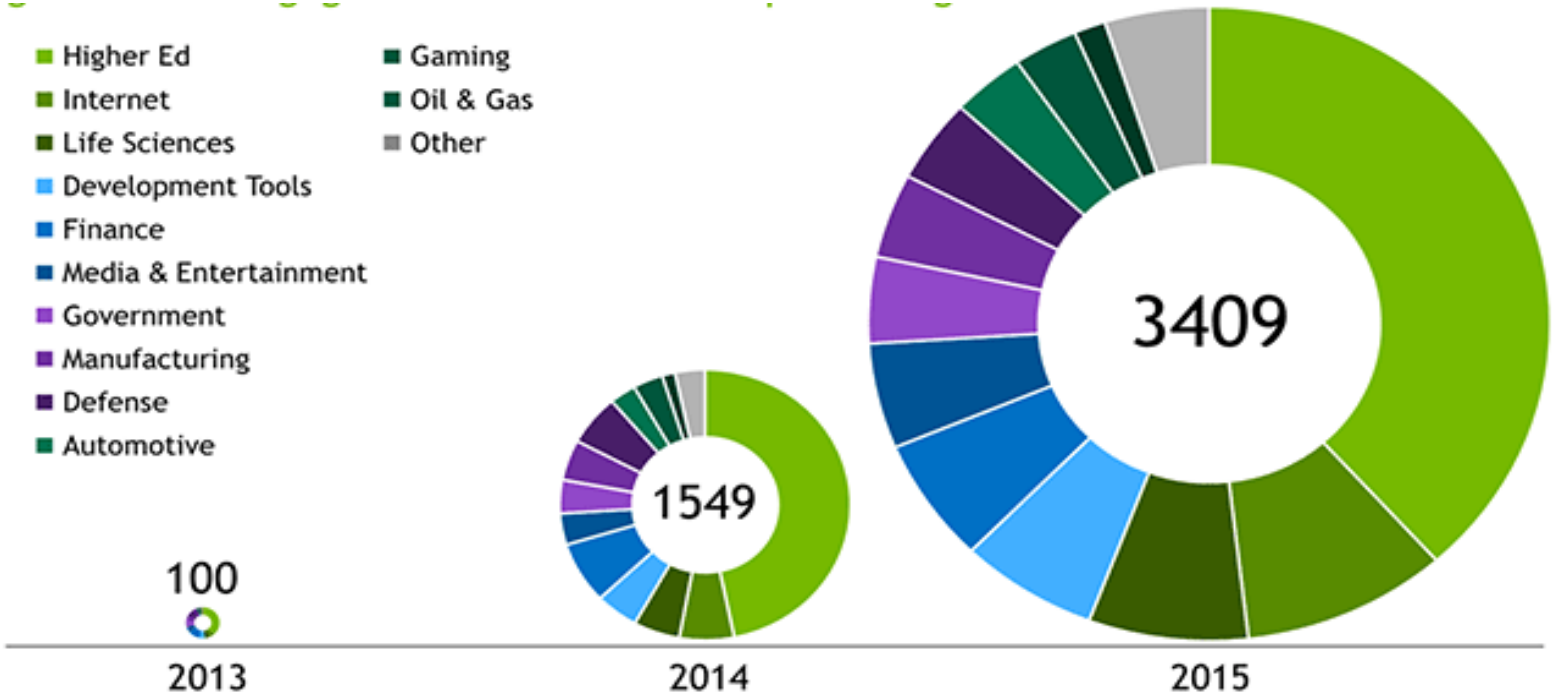
Real-time mobile management

The bottleneck in DOD's move t

NGA's supply chain for GEOINT a

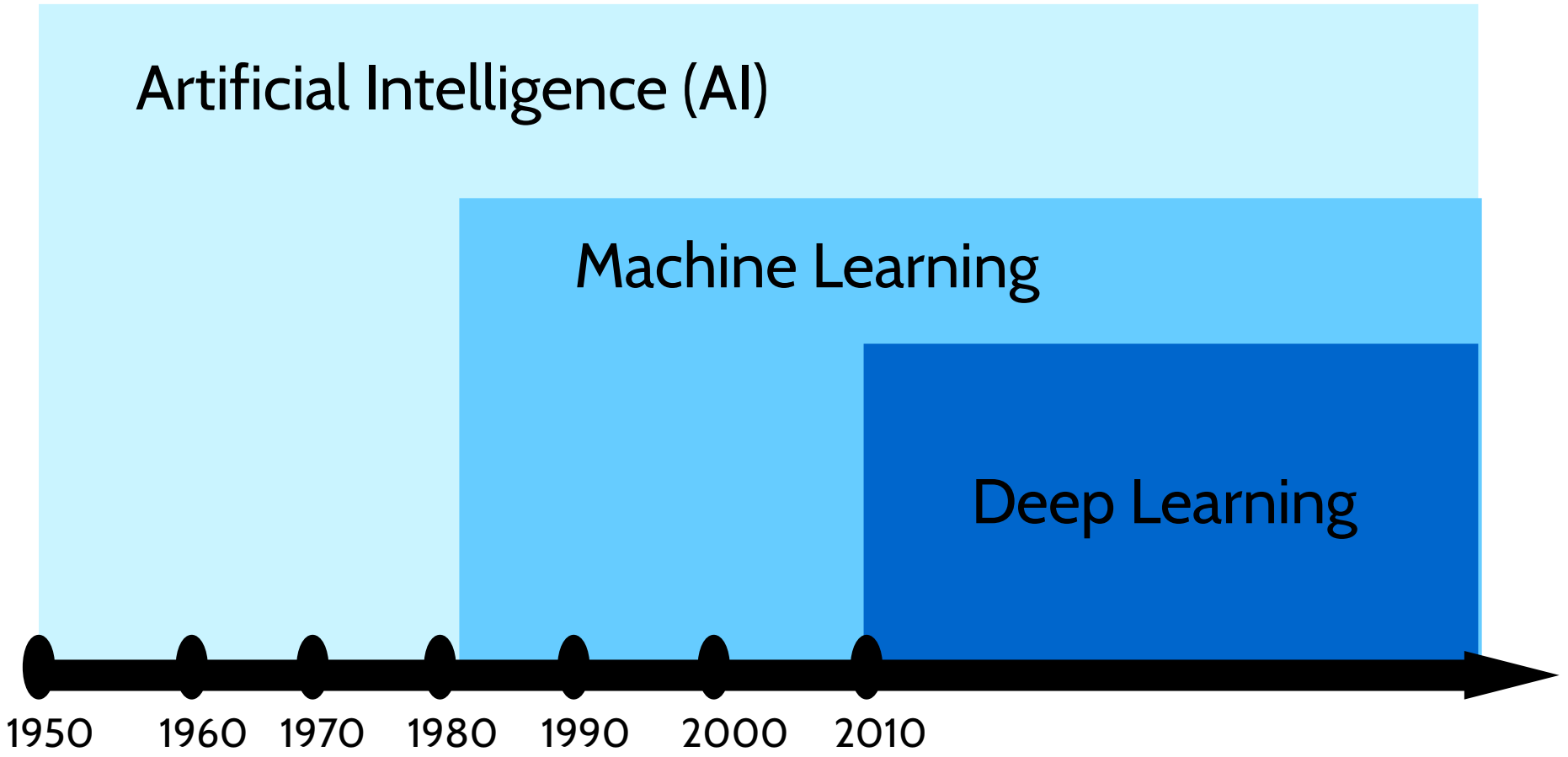
Industry and AI

Organizations engaged with NVIDIA on deep learning

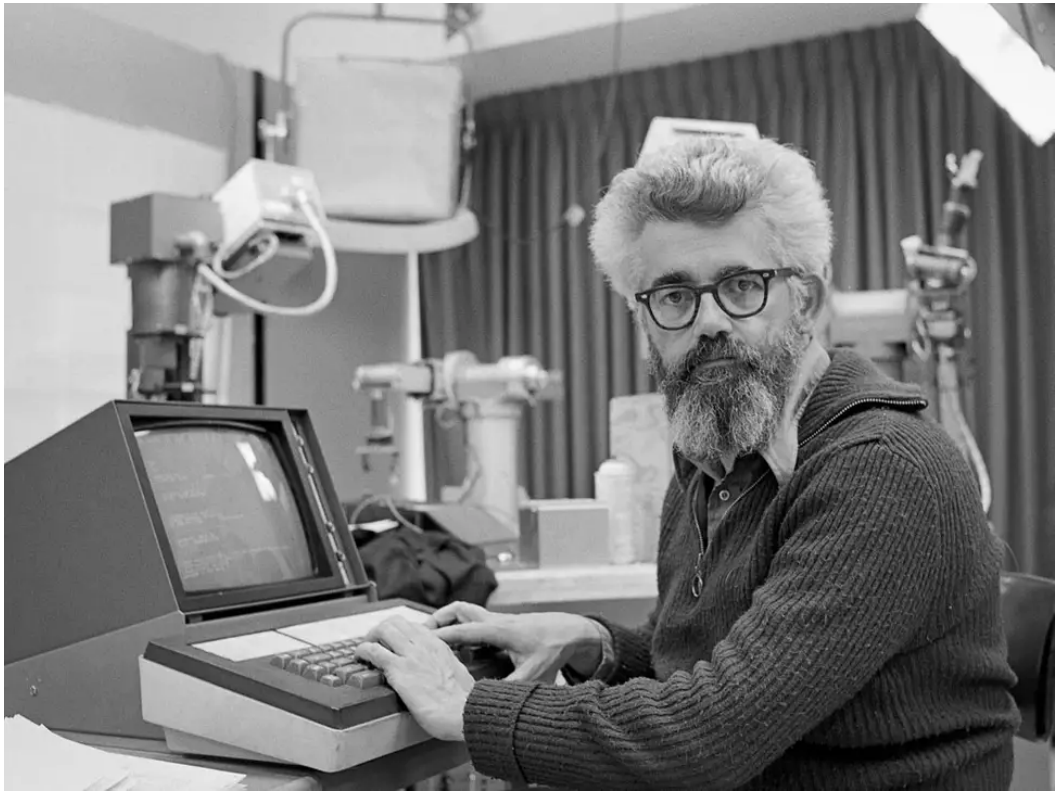


[NVIDIA Blog]

AI, Machine Learning & Deep Learning



Artificial Intelligence



“Our ultimate objective is to make programs that learn from their experience as effectively as humans do.”

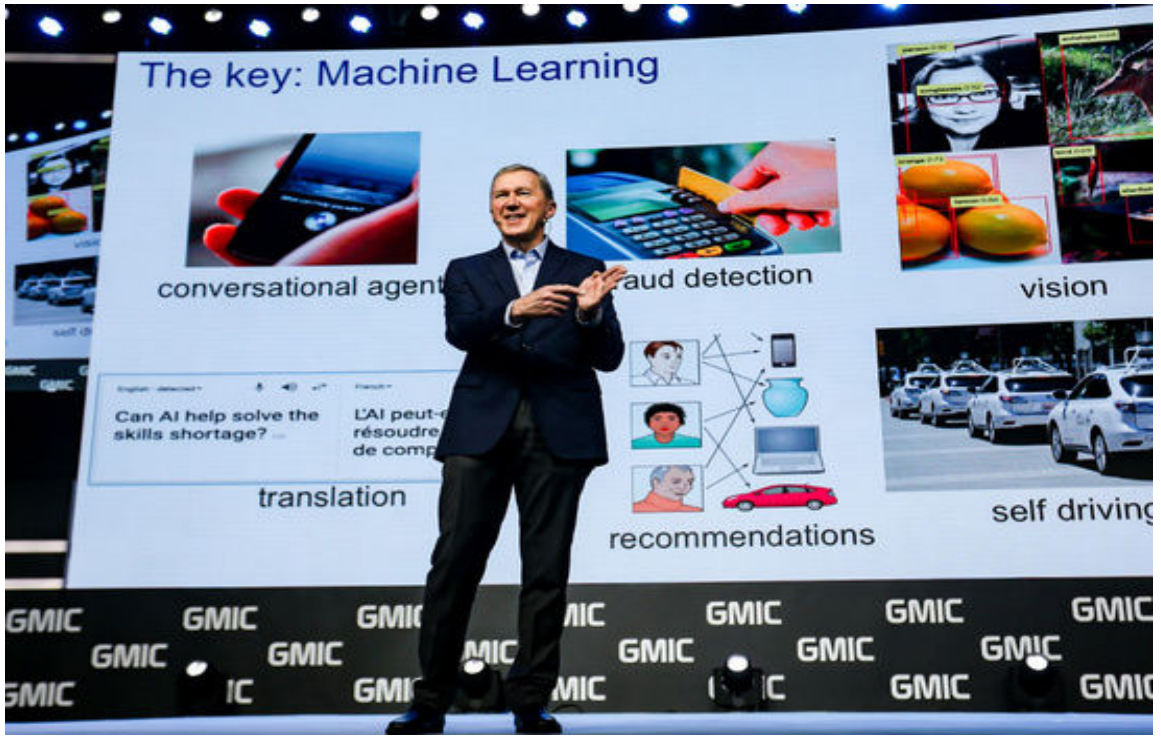
John McCarthy, 1958

Machine Learning

“Machine Learning is the science of getting computers to act without being explicitly programmed”

Andrew Ng

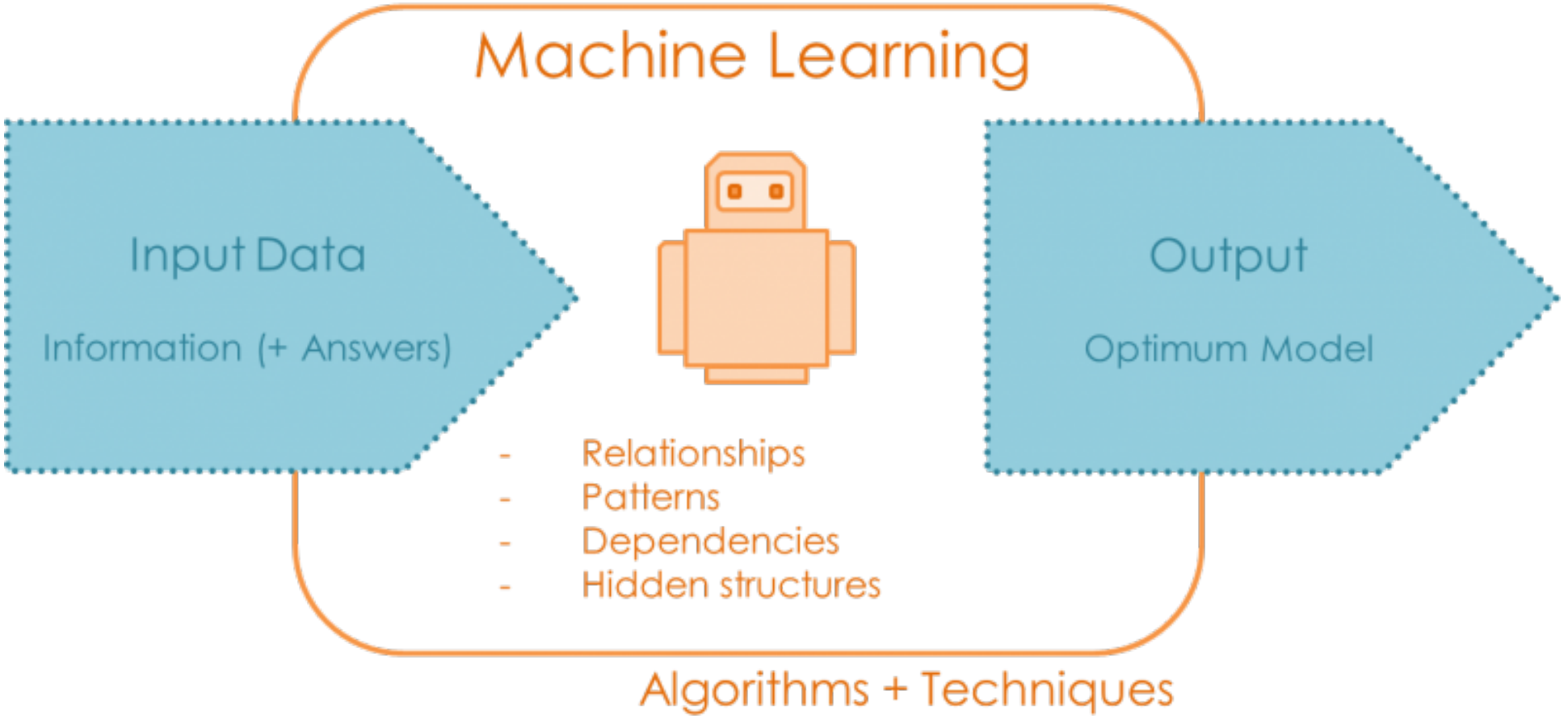
Machine Learning



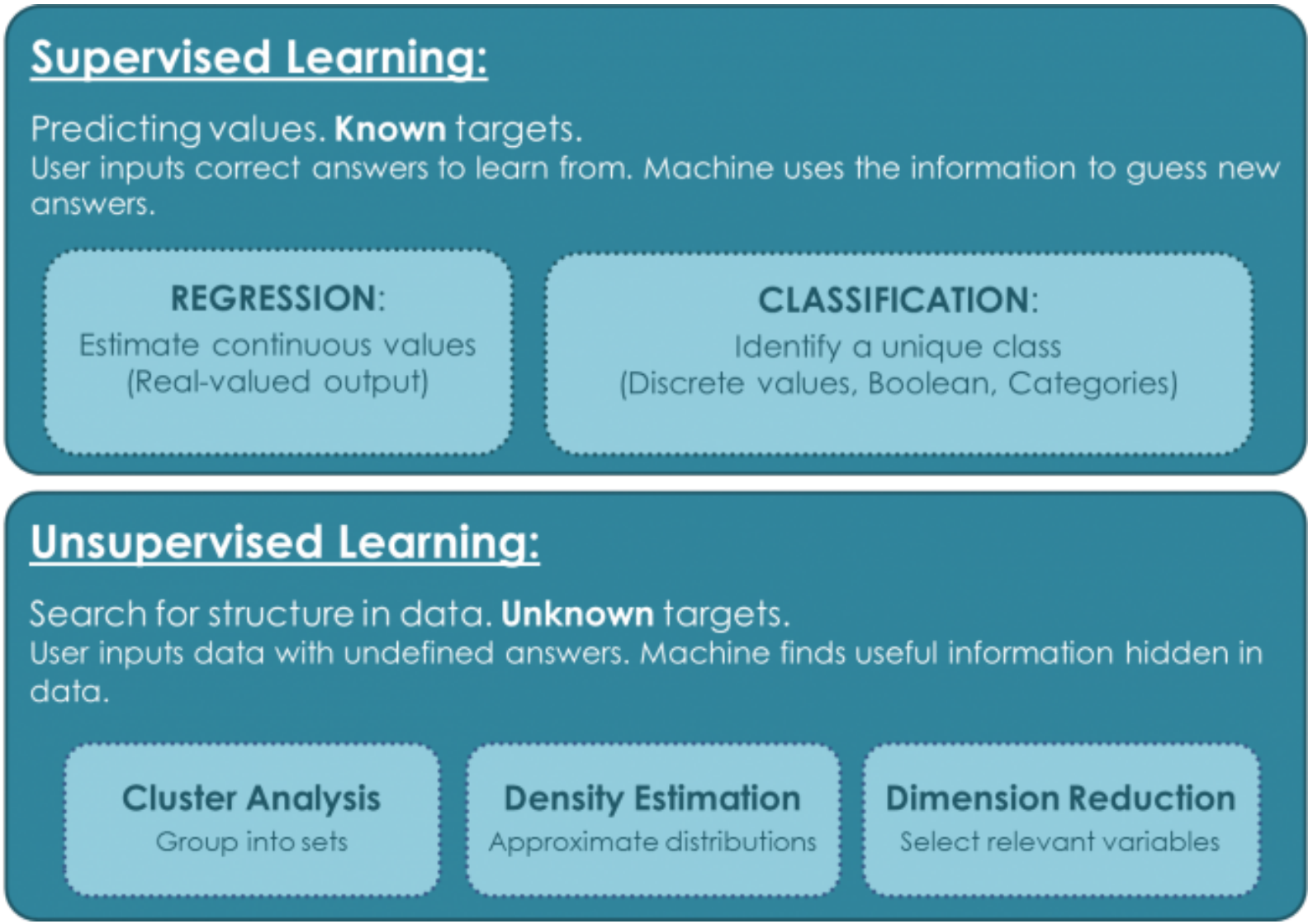
"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at task in T, as measured by P, improves with experience E."

Tom Mitchell, 1997

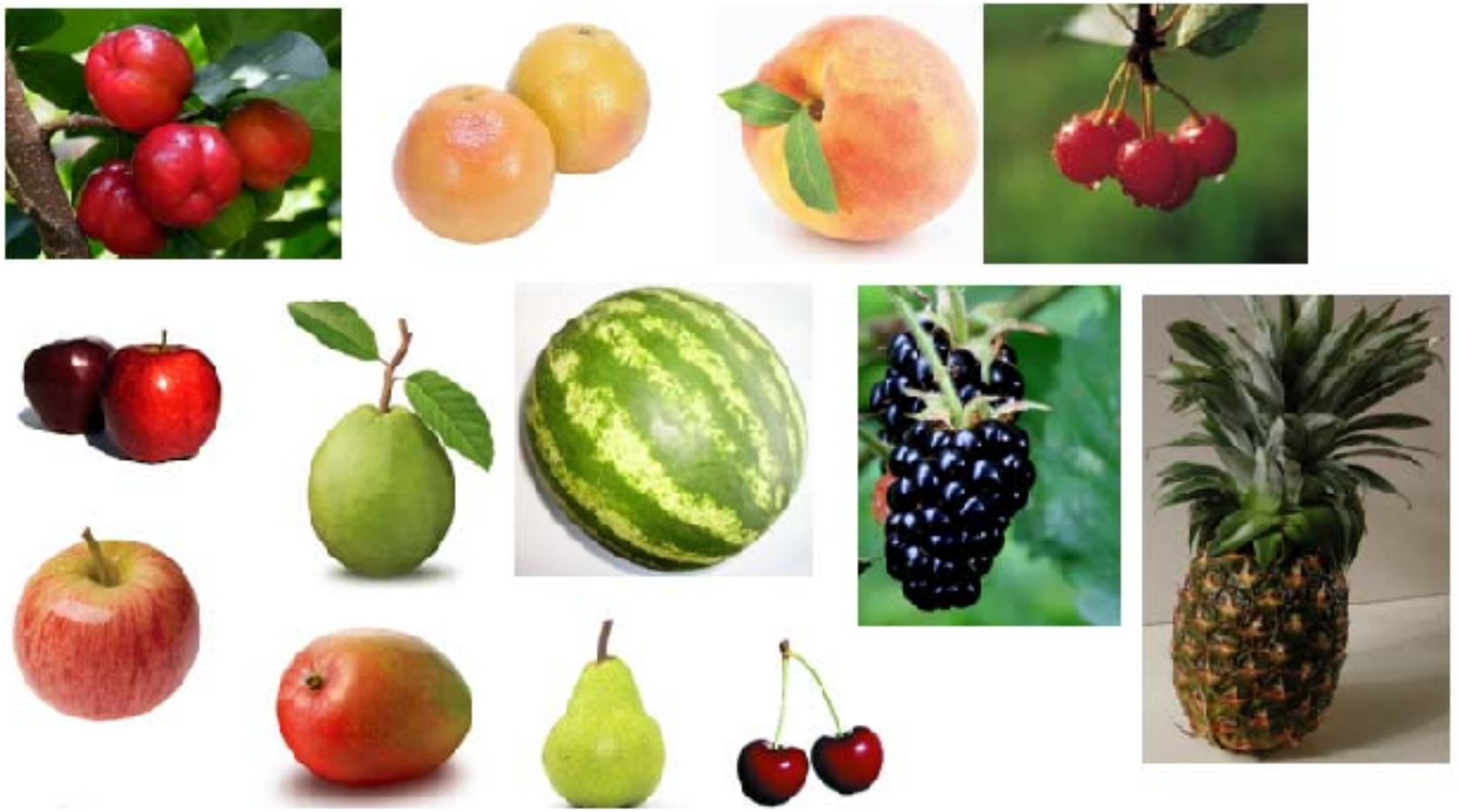
Machine Learning



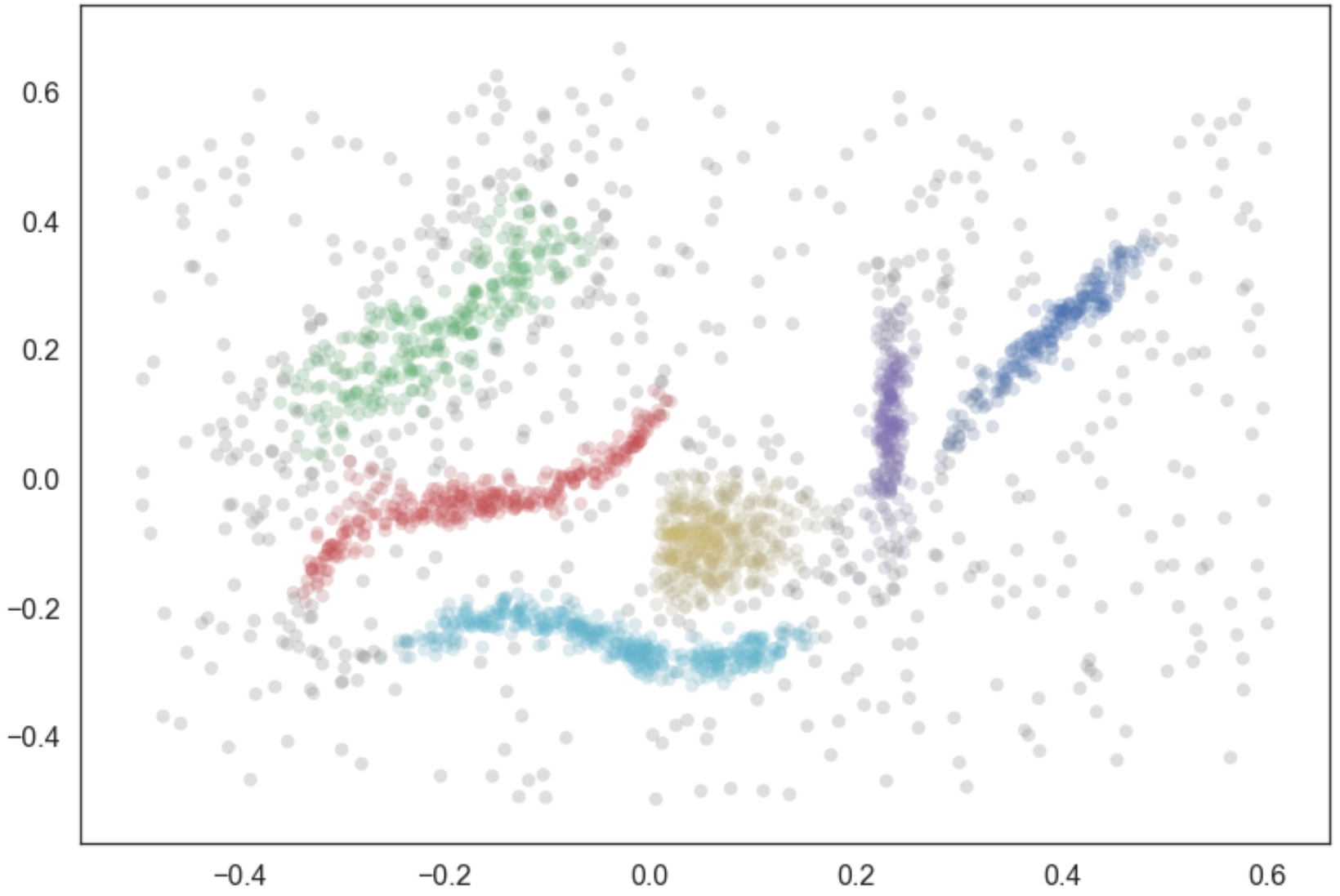
Machine Learning



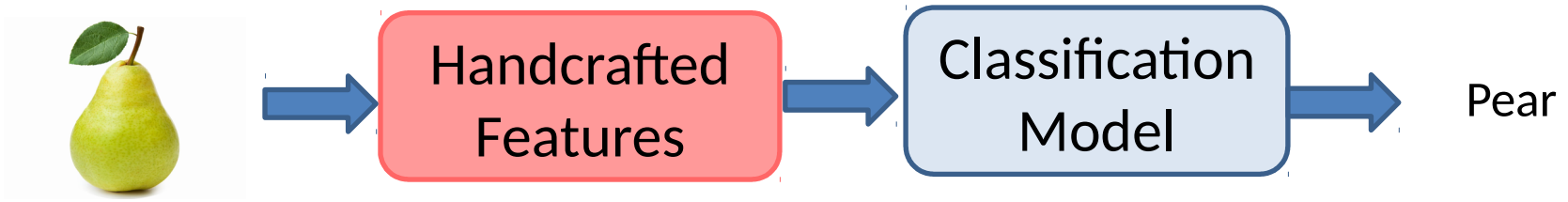
Raw Data vs Features



Raw Data vs Features

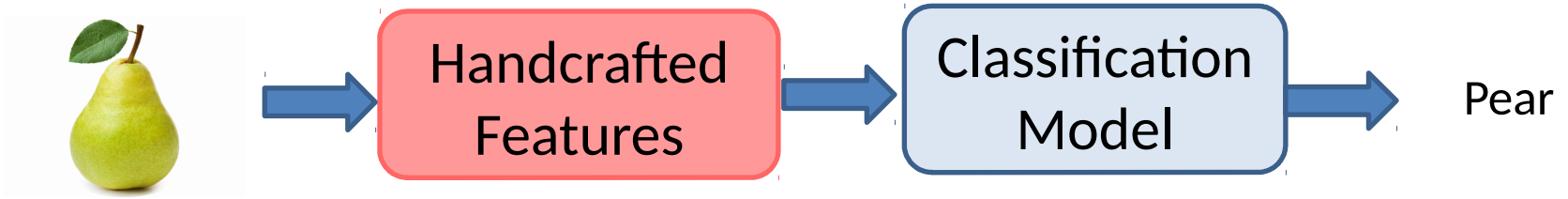


Raw Data vs Features

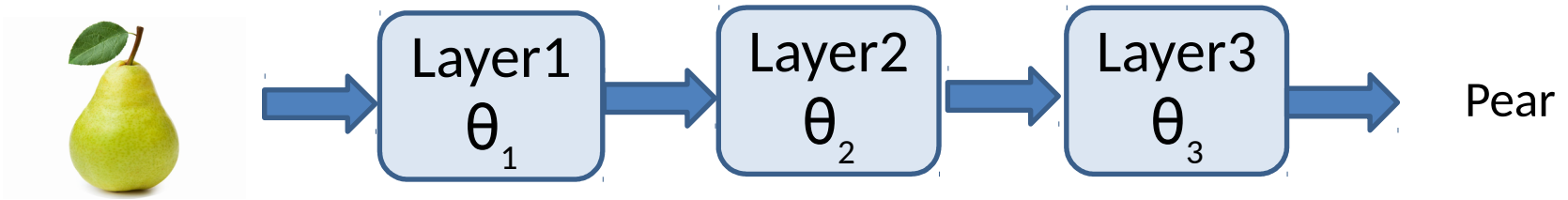


Learning hierarchical representations

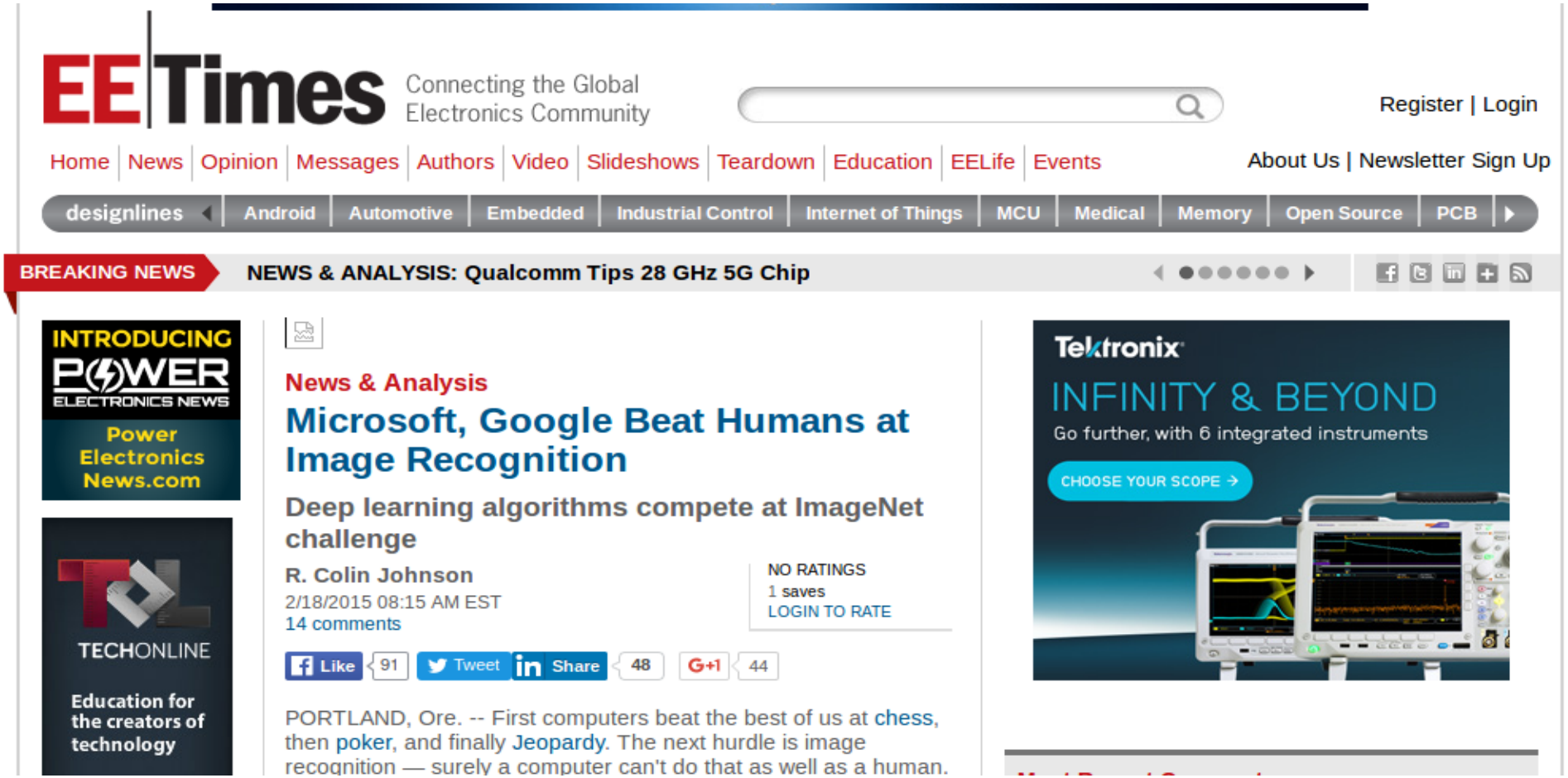
- Traditional framework



- Deep Learning



Visual Recognition



Machine Translation

Rick Rashid in **Tianjin, China**, October, 25, 2012

A voice recognition program translated a speech given by R. F. Rashid, Microsoft's top scientist, into Mandarin Chinese.

Creativity

PRISMA

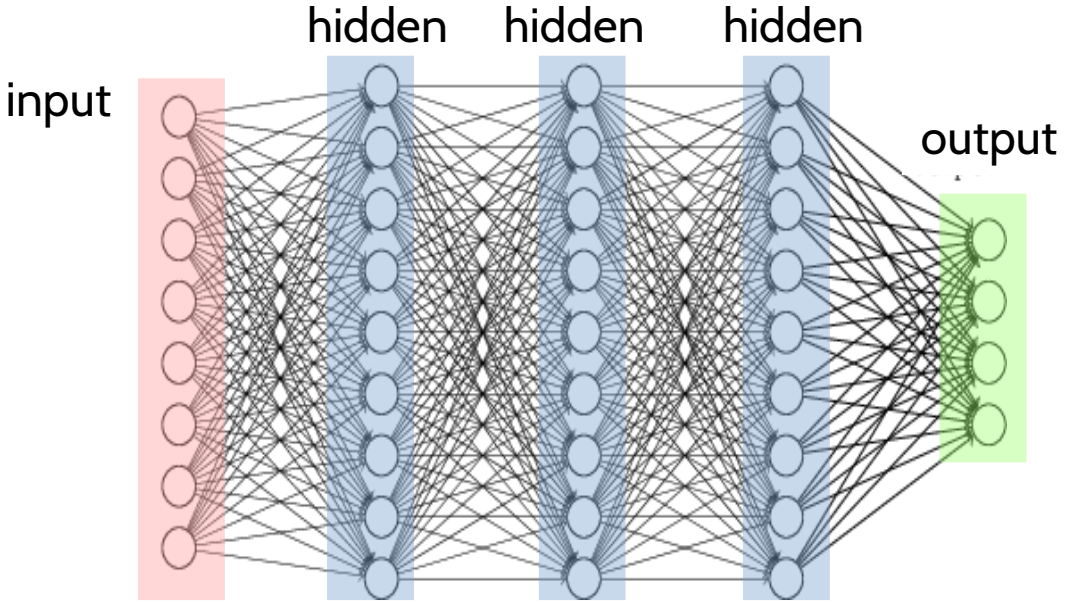
Deep Learning

- **What** is deep learning?
- **Why** is it generally **better than traditional ML** methods on image, speech and certain other types of data?

Deep Learning

- **What** is deep learning?

*Deep Learning means using a **neural network** with **several layers of nodes** between input and output*



More formally

- A family of **parametric** models which learn **non-linear hierarchical** representations:

$$a_L(\mathbf{x}; \boldsymbol{\Theta}) = h_L(h_{L-1}(...(h_1(\mathbf{x}, \boldsymbol{\theta}_1), \boldsymbol{\theta}_{L-1}), \boldsymbol{\theta}_L)$$

input

parameters of the network

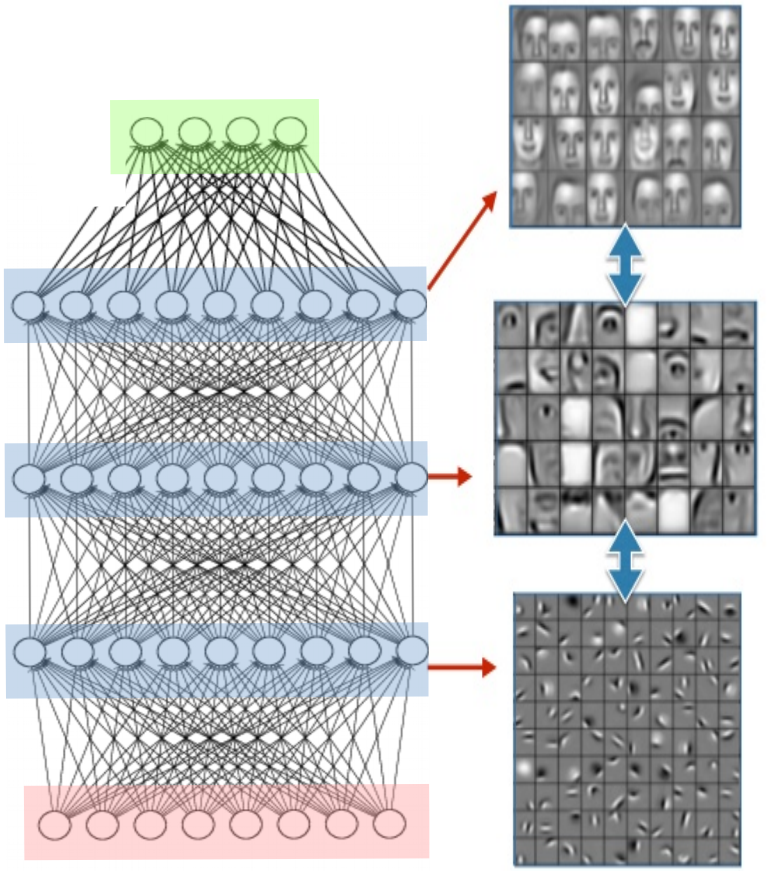
non-linear activation function

parameters of layer *L*

Deep Learning

- **Why** is it generally **better than other ML** methods on image, speech and certain other types of data?

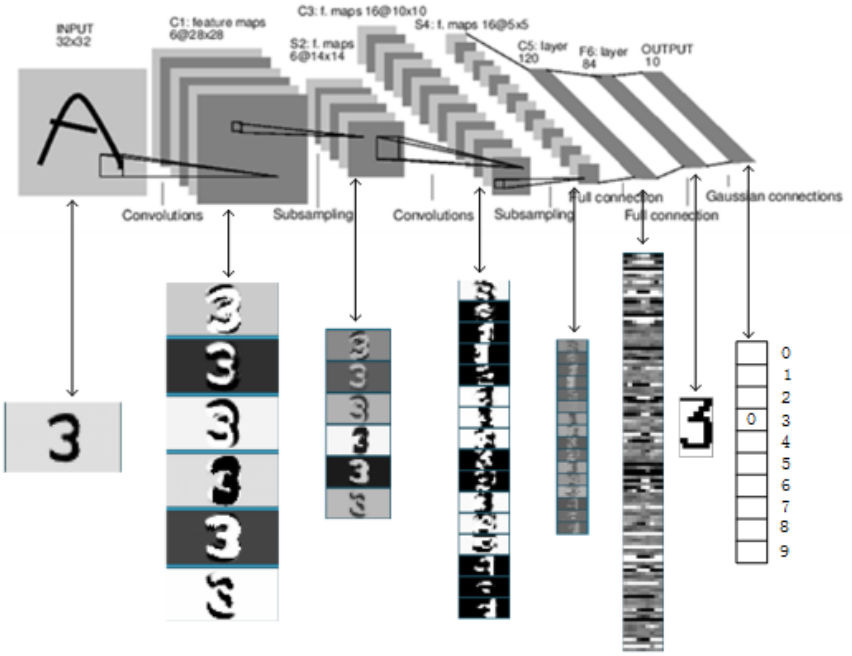
The series of layers between input and output compute ***relevant features automatically*** *in a series of stages, just as our brains seem to.*



Deep Learning

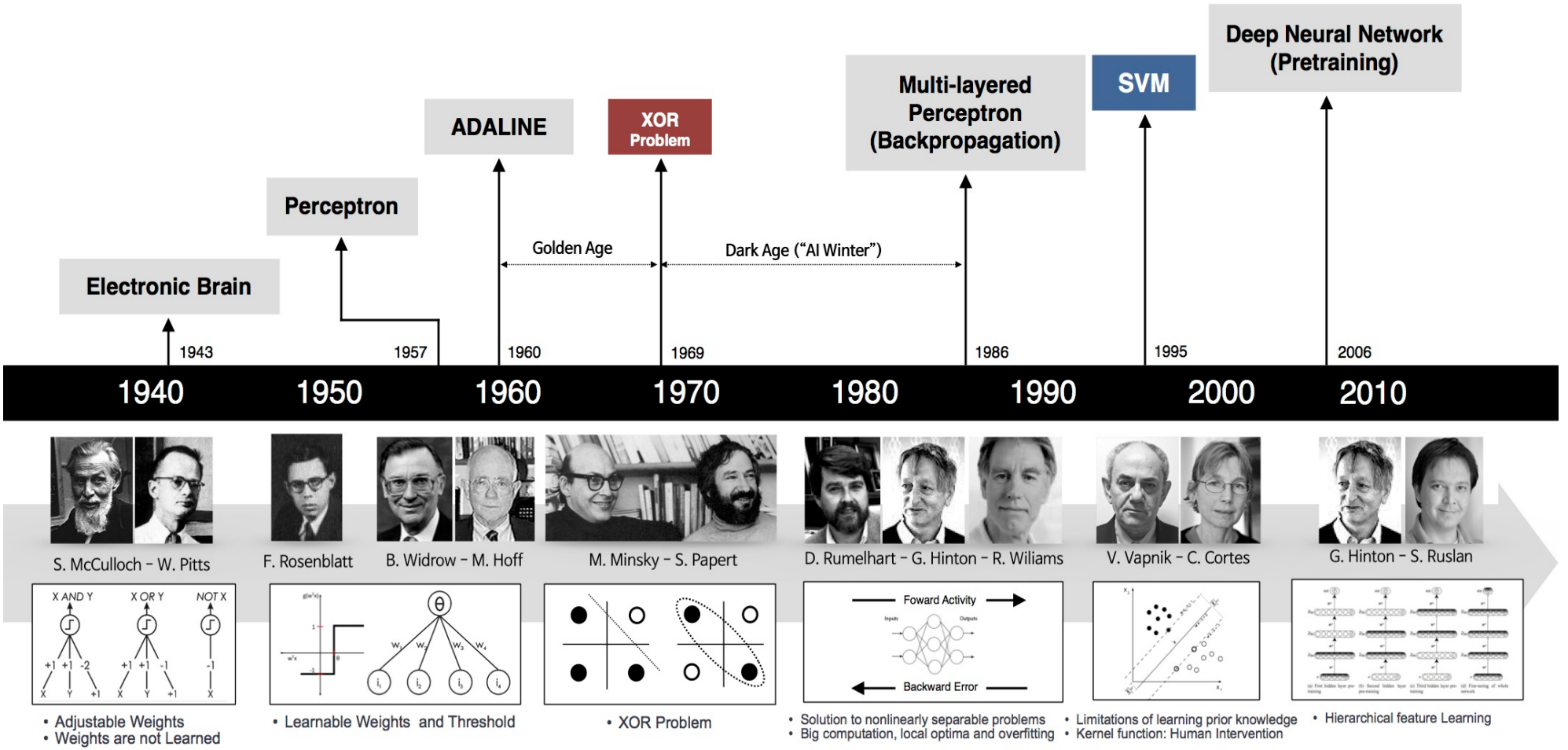
...but neural networks have been around for many years.

So, what is new?



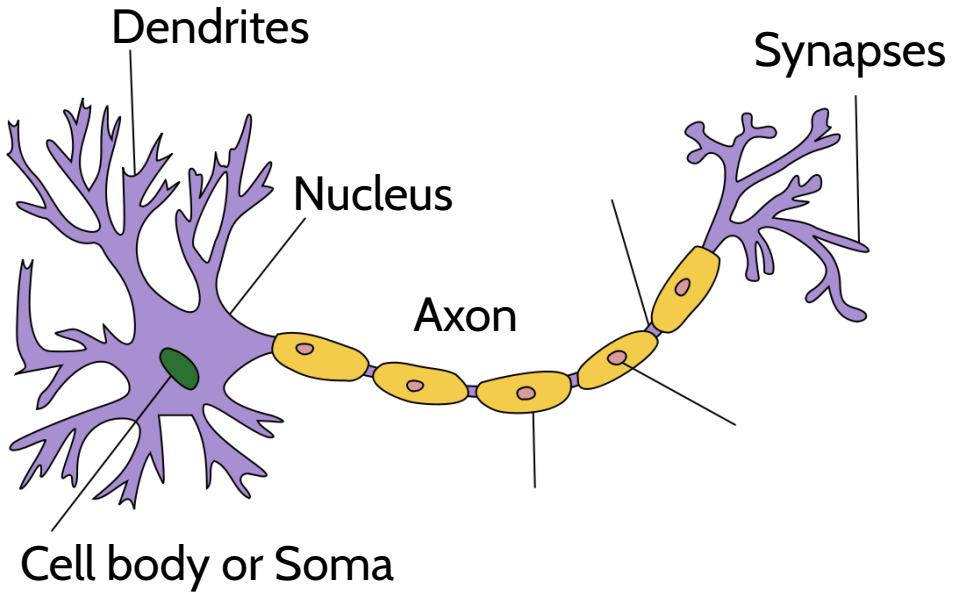
[LeCun, 1989]

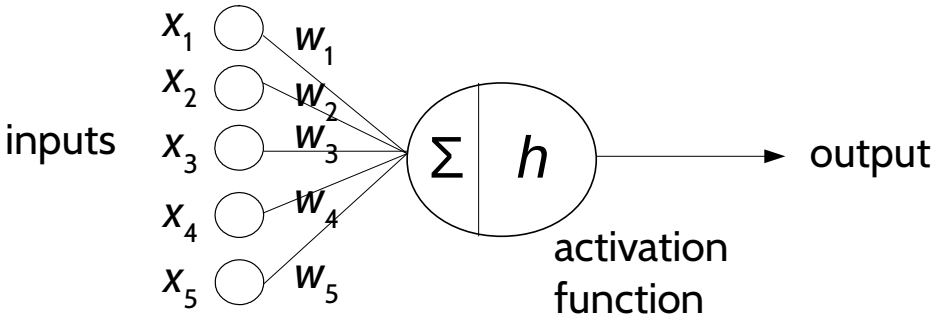
Brief History of Neural Networks



[VUNO]

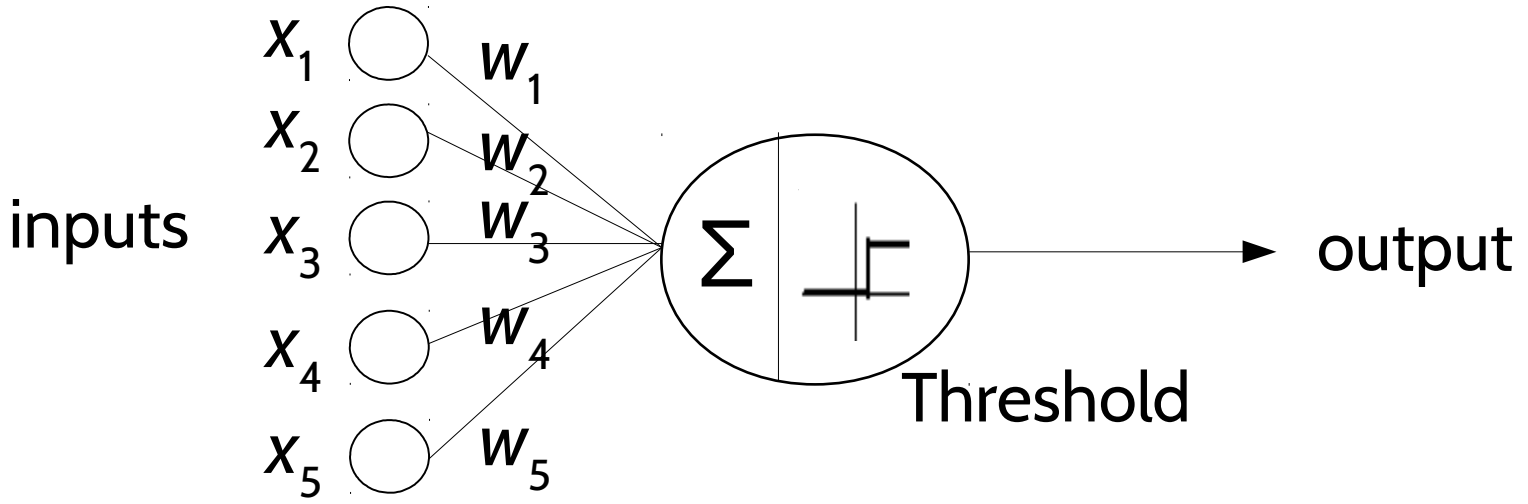
Biological neuron and Perceptron





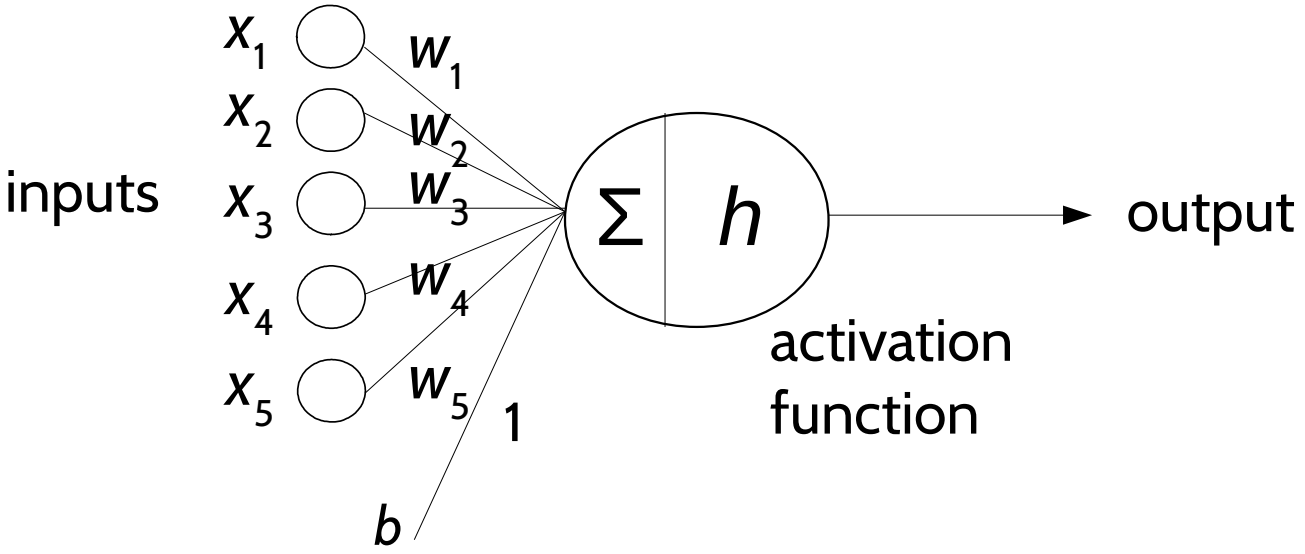
1943 – McCulloch & Pitts Model

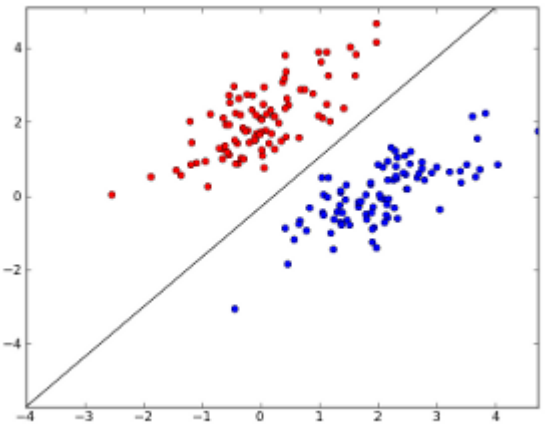
- Early model of artificial neuron
- Generates a binary output
- The weights values are fixed



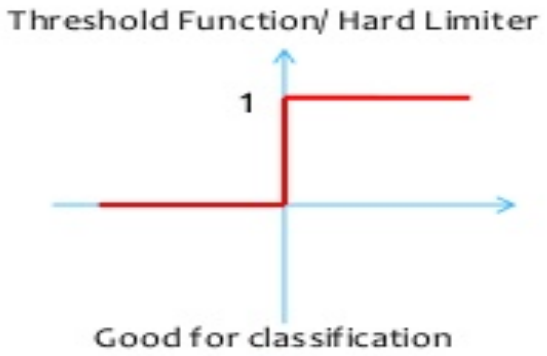
1958 – Perceptron by Rosemblatt

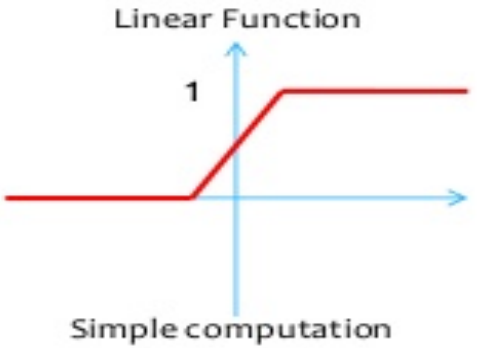
- Perceptron as a machine for linear classification
- Main idea: Learn the weights and consider bias.
  - One weight per input
  - Multiply weights with respective inputs and add bias
  - If result larger than **threshold** return 1, otherwise 0

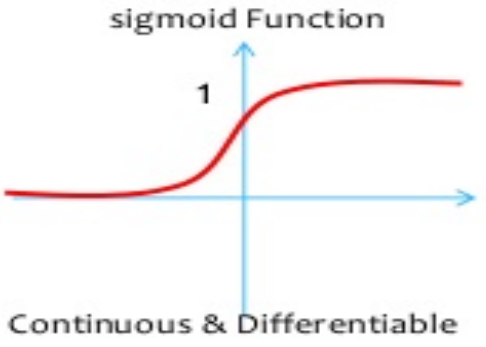




Activation functions



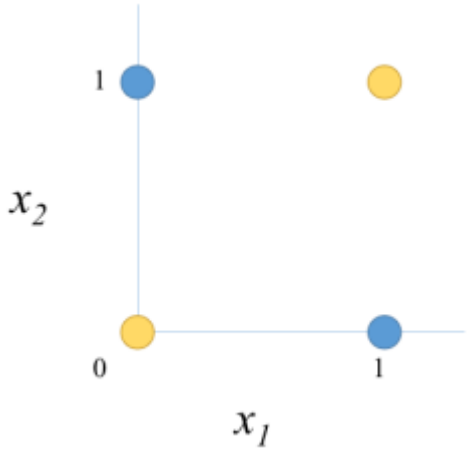




$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

First AI winter

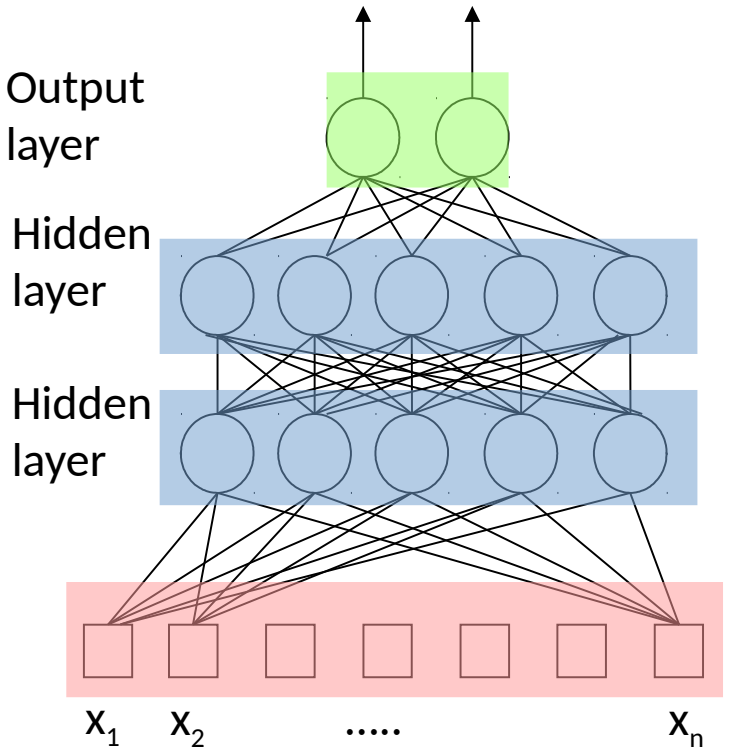
- 1970- Minsky. The XOR cannot be solved by perceptrons.
- Neural models cannot be applied to complex tasks.



Multi-layer Feed Forward Neural Network

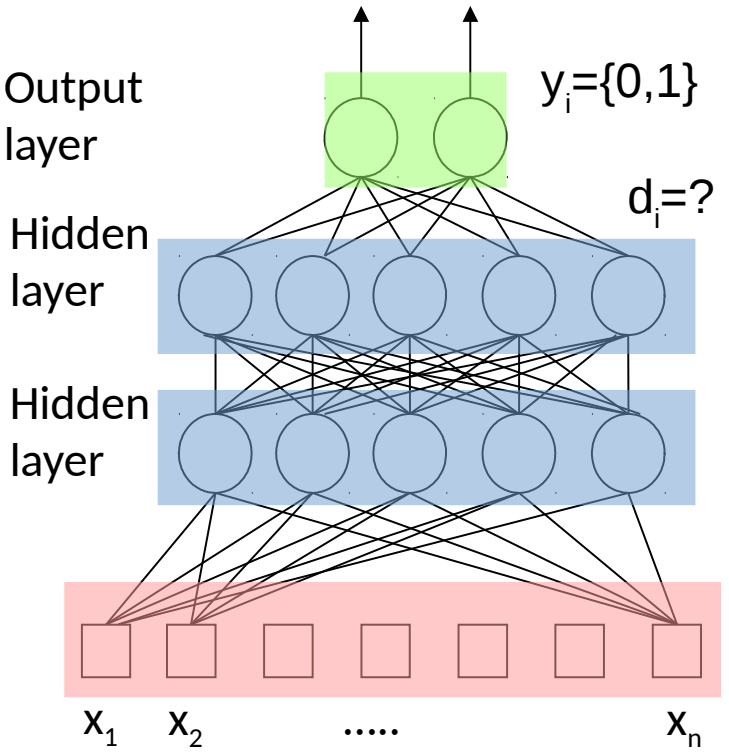
- 1980s. Multi-layer Perceptrons (MLP) can solve XOR.

- ML Feed Forward Neural Networks:
  - Densely connect artificial neurons to realize compositions of non-linear functions
  - The information is propagated from the inputs to the outputs
  - The input data are usually *n*-dimensional feature vectors
  - Tasks: Classification, Regression



How to train a MLP?

- Rosenblatt's algorithm not applicable, as it expects to know the desired target.
- For hidden layers we cannot know the desired target



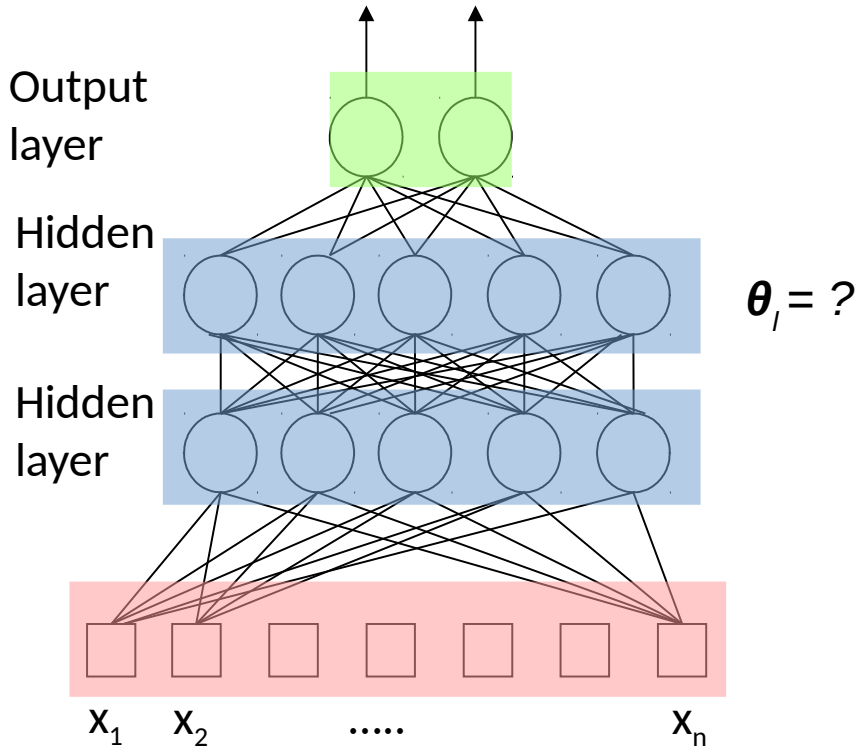
1986 – Backpropagation

- Learning MLP for complicated functions can be solved.
- Backpropagation: efficient algorithm for complex NN which processes “large” training sets.
- Today backpropagation is still at the core of NN training.

Werbos (1974). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. Thesis, Harvard University.
Rumelhart, Hintont, Williams (1986). Learning representations by back-propagating errors. Nature

Backpropagation

Learning is the process of modifying the weights of each layer $\boldsymbol{\theta}_l$ in order to produce a network that performs some function:



Backpropagation

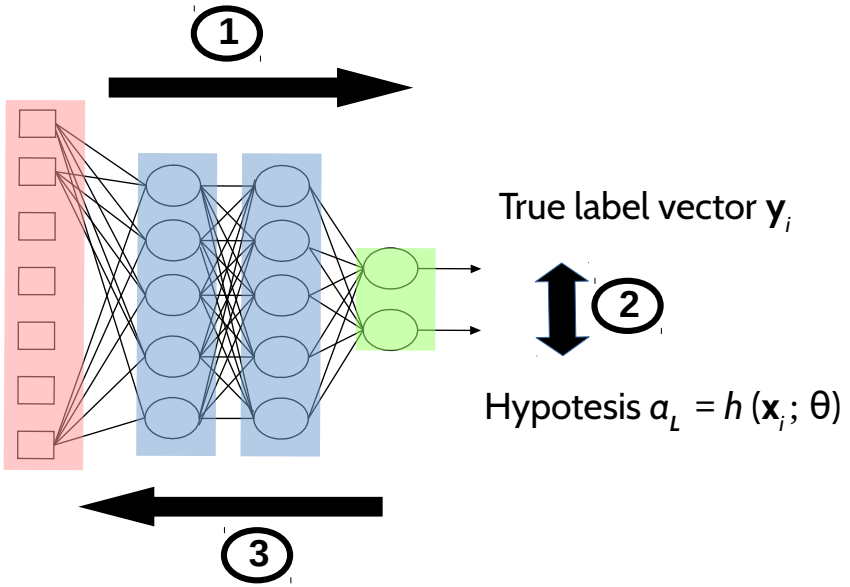
- Preliminary steps:
  - Collect a training set $\{x_i, y_i\}$
  - Define model and initialize randomly weights.

- Given the training set find the weights:

$$a_L(\mathbf{x}; \boldsymbol{\Theta}) = h_L(h_{L-1}(...(h_1(\mathbf{x}, \boldsymbol{\theta}_1), \boldsymbol{\theta}_{L-1}), \boldsymbol{\theta}_L)$$

$$\boldsymbol{\Theta}^* = \arg\min_{\boldsymbol{\Theta}} \sum_{\mathbf{x}_i, y_i} \ell\left(y_i, a_L(\mathbf{x}_i; \boldsymbol{\Theta})\right)$$

Backpropagation




```
Randomly initialize the weights

WHILE error is too large
     (1) For each training sample (presented in random order)
     Apply the inputs to the network
     Calculate the output for every neuron from the input layer, through the
       hidden layers, to the output layer
     (2) Calculate the error at the outputs
     (3) Use the output error to compute error signals for previous layers
     Use the error signals to compute weight adjustments
     Apply the weight adjustments

Periodically evaluate the network performance
```

Backpropagation

- Optimization with gradient descent:

$$\Theta^* = \arg\min_{\Theta} \sum_{\mathbf{x}_i, y_i} \ell\left(y_i, a_L(\mathbf{x}_i; \Theta)\right)$$

$$\Theta^{t+1} = \Theta^t - \eta_t \nabla_\Theta \mathcal{L}$$

  - The most important component is how to compute the gradient
  - The backward computations of network return the gradient
  - Efficient due to recursive computations

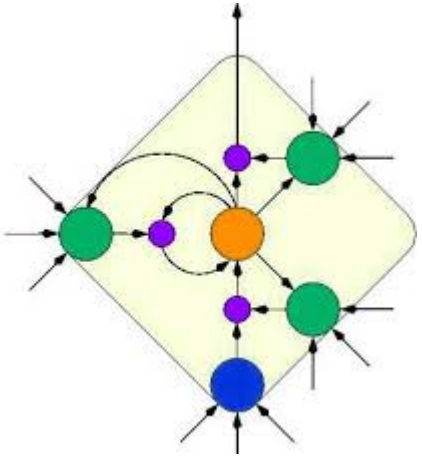
$$\frac{\partial \mathcal{L}}{\partial a_l} = \left(\frac{\partial a_{l+1}}{\partial x_{l+1}}\right)^T \cdot \frac{\partial \mathcal{L}}{\partial a_{l+1}} \quad \Rightarrow \quad \frac{\partial \mathcal{L}}{\partial \theta_l} = \frac{\partial a_l}{\partial \theta_l} \cdot \left(\frac{\partial \mathcal{L}}{\partial a_l}\right)^T$$

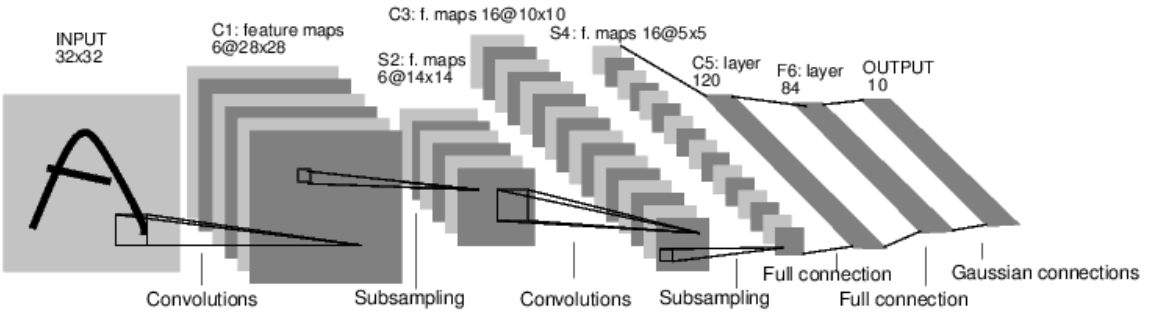
Recursive rule: Previous layer

Current layer

1990s - CNN and LSTM

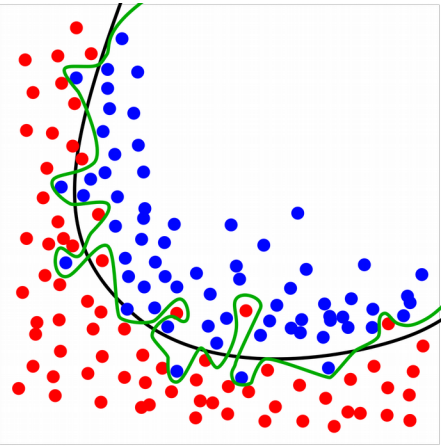
- Important advances in the field:
  - Backpropagation
  - Recurrent Long-Short Term Memory Networks (Schmidhuber, 1997)
  - Convolutional Neural Networks - LeNet: OCR solved before 2000s (LeCun, 1998).

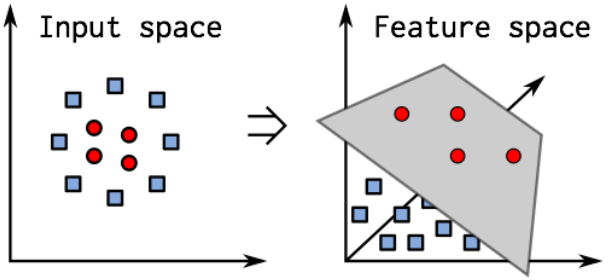




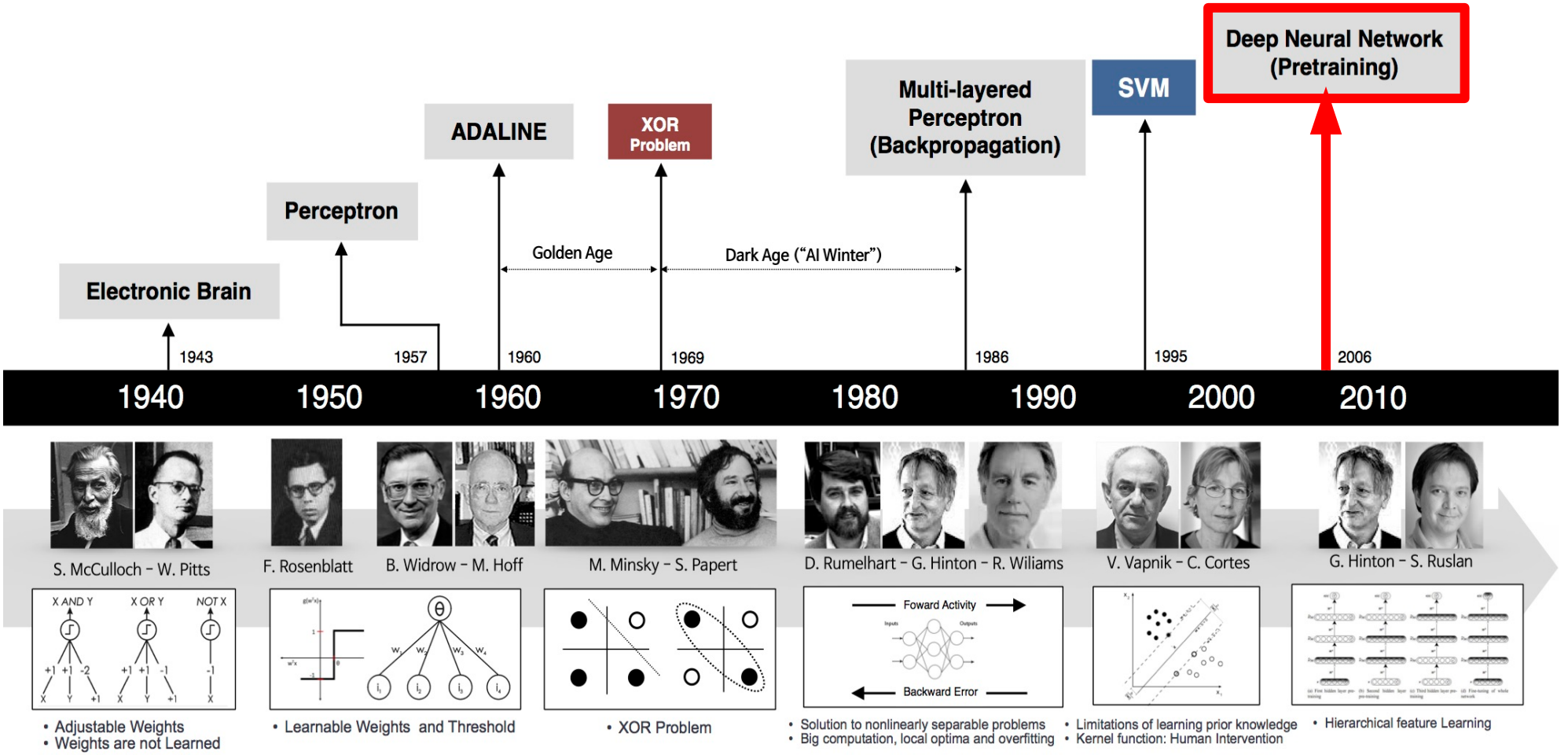
Second AI winter

- NN cannot exploit many layers
  - Overfitting
  - Vanishing gradient (with NN training you need to multiply several small numbers → they become smaller and smaller)
- Lack of processing power (no GPUs)
- Lack of data (no large annotated datasets)
- Kernel Machines (e.g. SVMs) suddenly become very popular◦





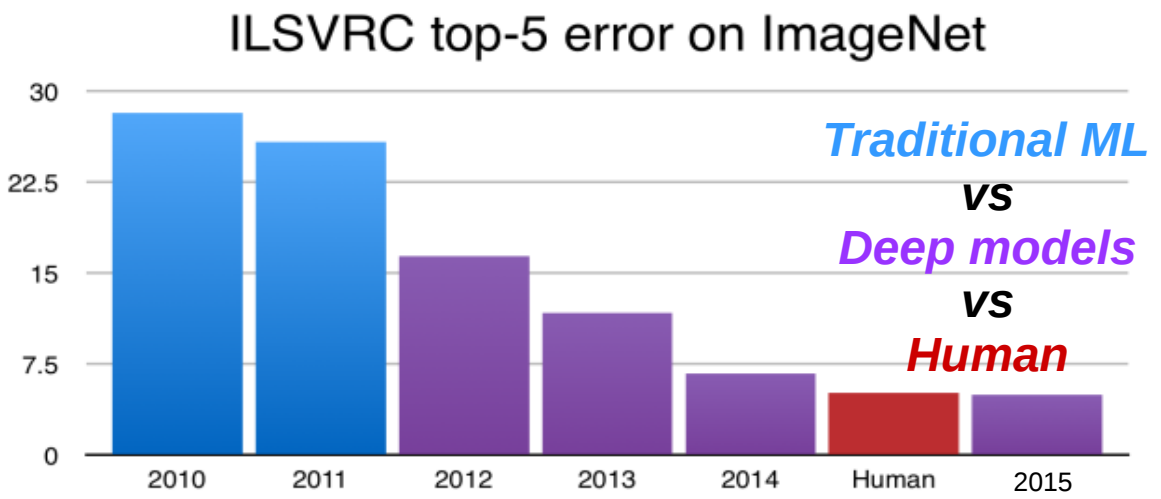
Pretraining of Deep Neural Networks



[VUNO]

2012 - AlexNet

- Hinton's group implemented a CNN similar to LeNet [LeCun1998] but...
  - Trained on ImageNet (1.4M images, 1K categories)
  - With 2 GPUs
  - Other technical improvements (ReLU, dropout, data augmentation)

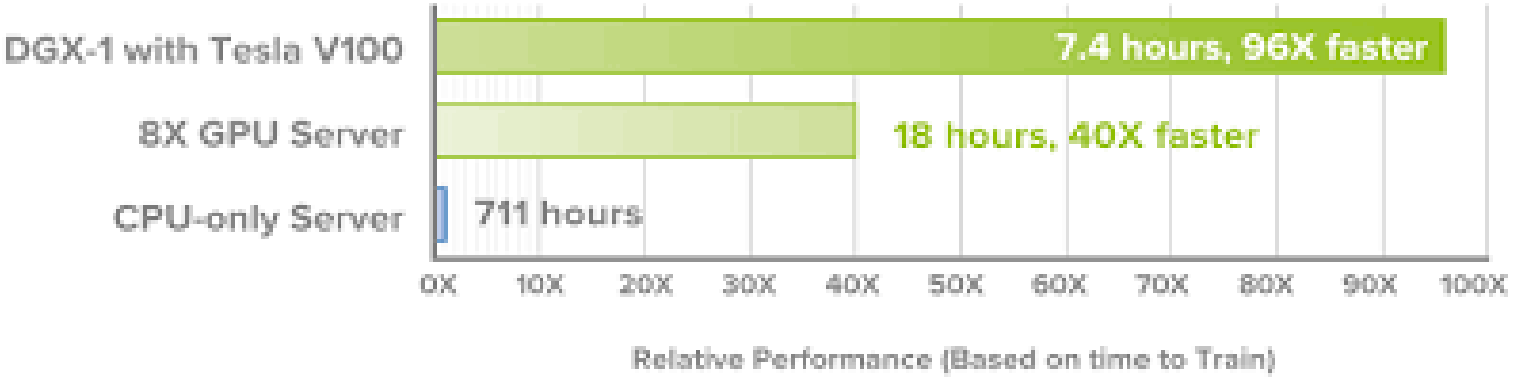


Why Deep Learning now?

- Three main factors:
  - Better hardware
  - Big data
  - Technical advances:
    - Layer-wise pretraining
    - Optimization (e.g. Adam, batch normalization)
    - Regularization (e.g. dropout)

      ....

GPUs

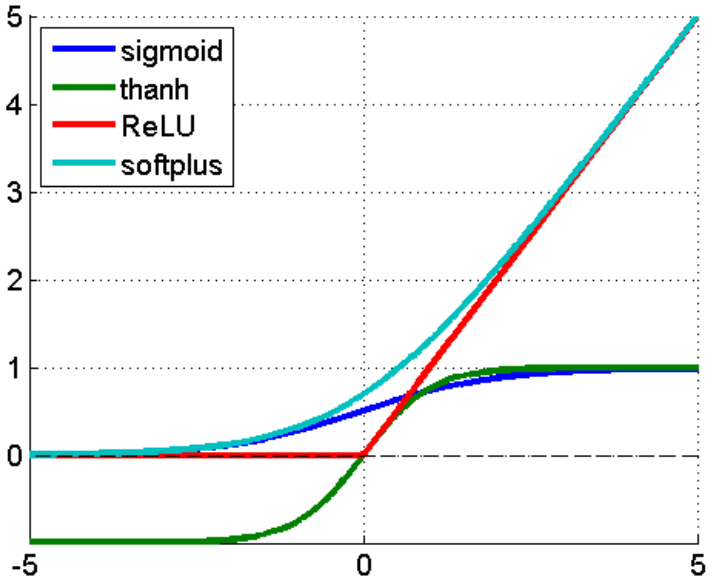


[NVIDIA Blog]

Large fully annotated datasets

Rectified Linear Units

$$f(x) = \max(0, x)$$



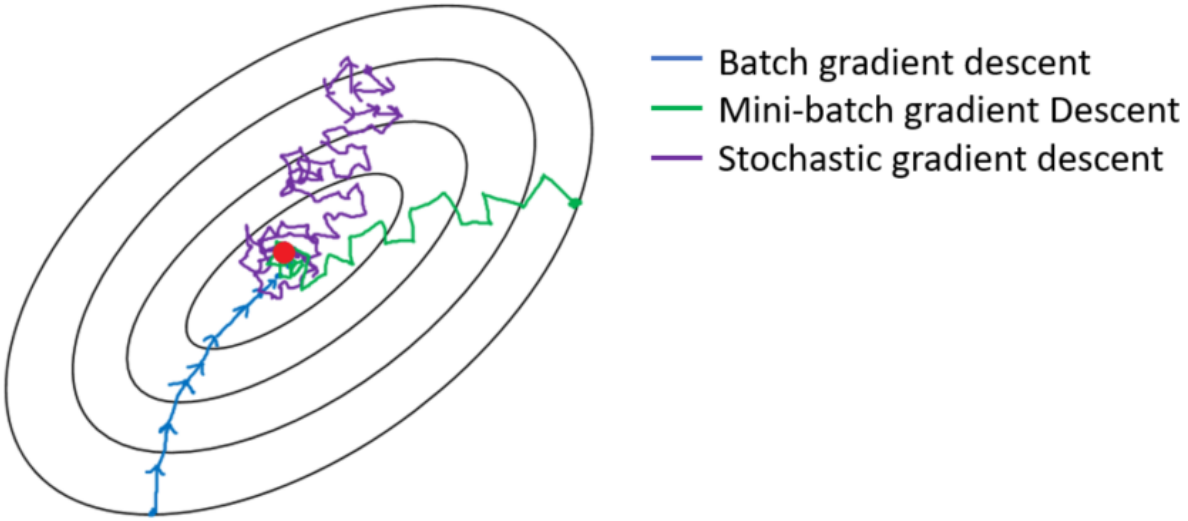
- More efficient gradient propagation:
  (derivative is 0 or constant)
- More efficient computation:
  (only comparison, addition and multiplication).
- Sparse activation: e.g. in a randomly initialized networks, only about 50% of hidden units are activated (having a non-zero output)
- Lots of variations have been proposed recently.

Stochastic Gradient Descent (SGD)

- Use mini-batch **sampled** in the dataset for gradient estimate.

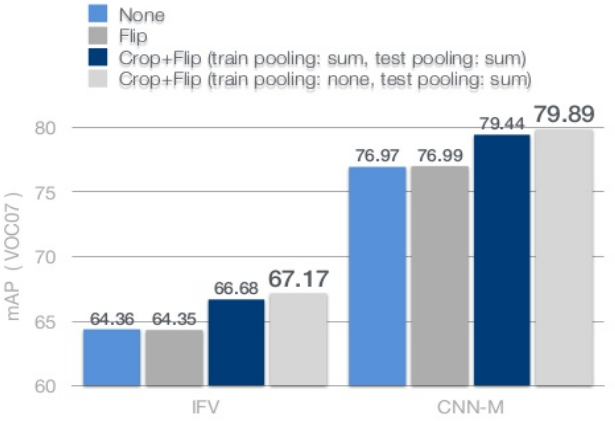
$$\Theta^{t+1} = \Theta^{t} - \frac{\eta_t}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \nabla_{\Theta} \mathcal{L}_i$$

- Sometimes helps to escape from local minima
- Noisy gradients act as regularization
- Variance of gradients increases when batch size decreases
- Not clear how many sample per batch



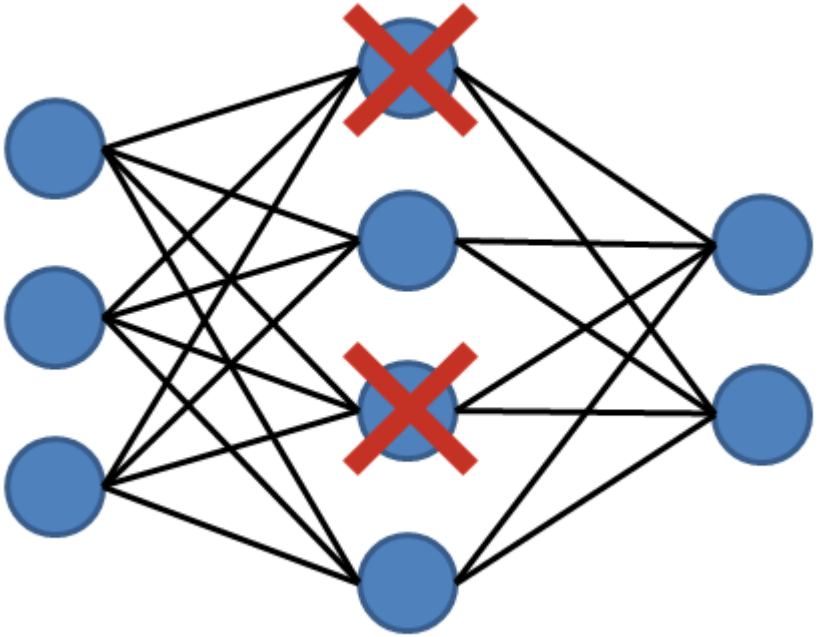
Data augmentation

- Simple preprocessing makes the difference (e.g. image flipping, scaling)



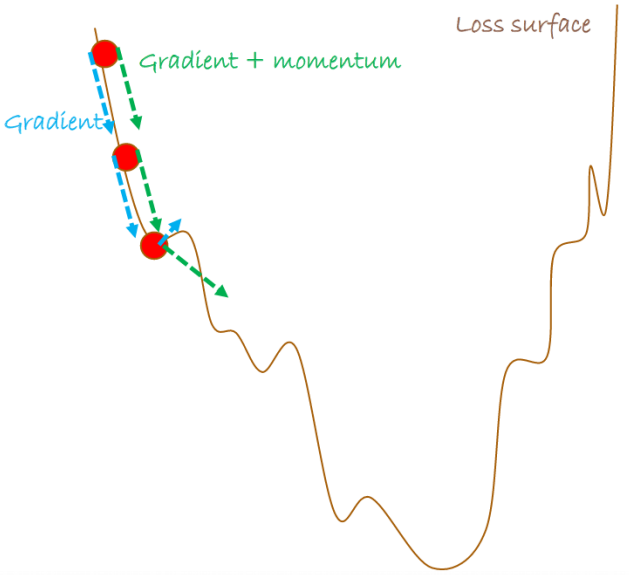
Regularization - Dropout

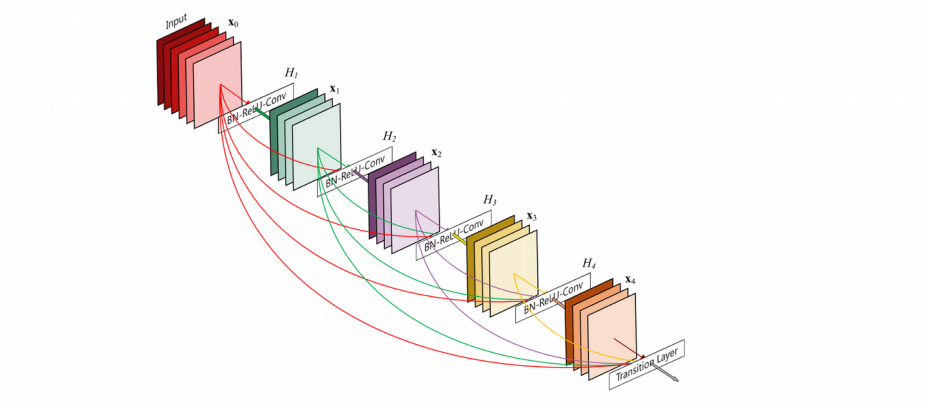
- For each instance drop a node (hidden or input) and its connections with probability *p* and train
- Final net just has all averaged weights (actually scaled by 1-*p*)
- As if ensembling 2^n different network substructures



Technical Advances

- Activation functions and losses (e.g. ReLU and cross-entropy)
- Data augmentation and pretraining
- Address overfitting (e.g. dropout)
- Training schemes (e.g. Adam)
- Architectures (e.g. ResNet, DenseNet…)
- Weights initialization
- Model distillation
- Batch normalization
- Deep domain adaptation
- ….





Batch Normalization

- Idea: renormalize activations.
- Obtain zero-mean and unit variance inputs. Scale and shift the normalized activation with two learnable weights γ and β:

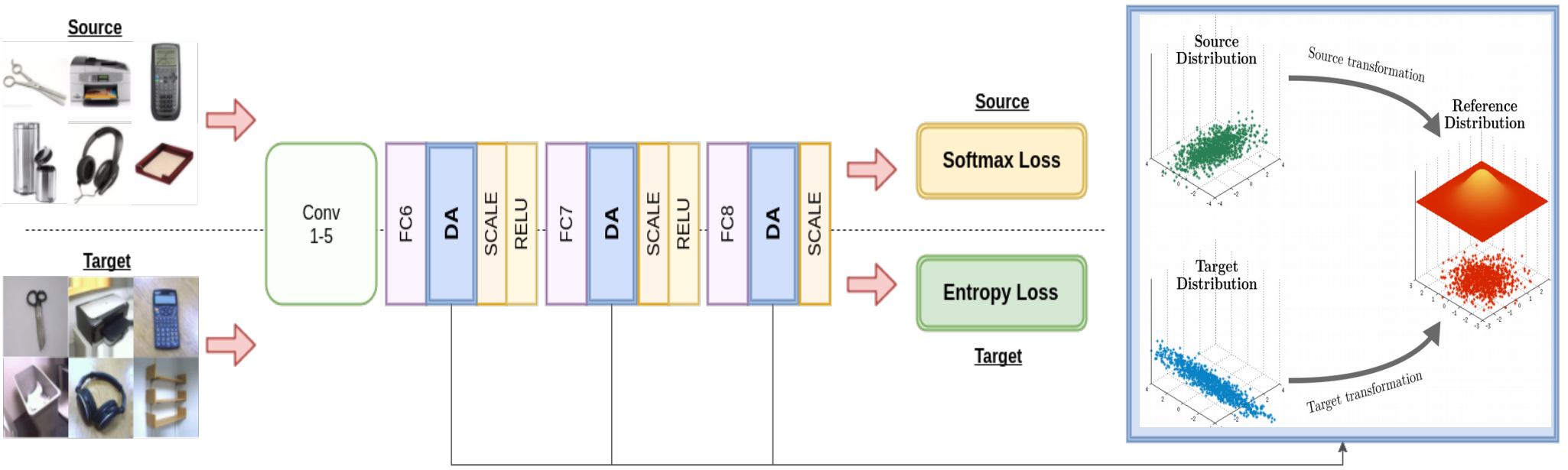
$$\widehat{x}^{(k)} = \frac{x^{(k)} - \mathrm{E}[x^{(k)}]}{\sqrt{\mathrm{Var}[x^{(k)}]}}$$

$$y^{(k)} = \gamma^{(k)}\widehat{x}^{(k)} + \beta^{(k)}$$

- Problem: Compute mean and variance of that activation for the entire data set. Solution: Approximate over a mini-batch.
- BN advantages: allows larger learning rates, improves gradient flow, reduces dependence on initialization

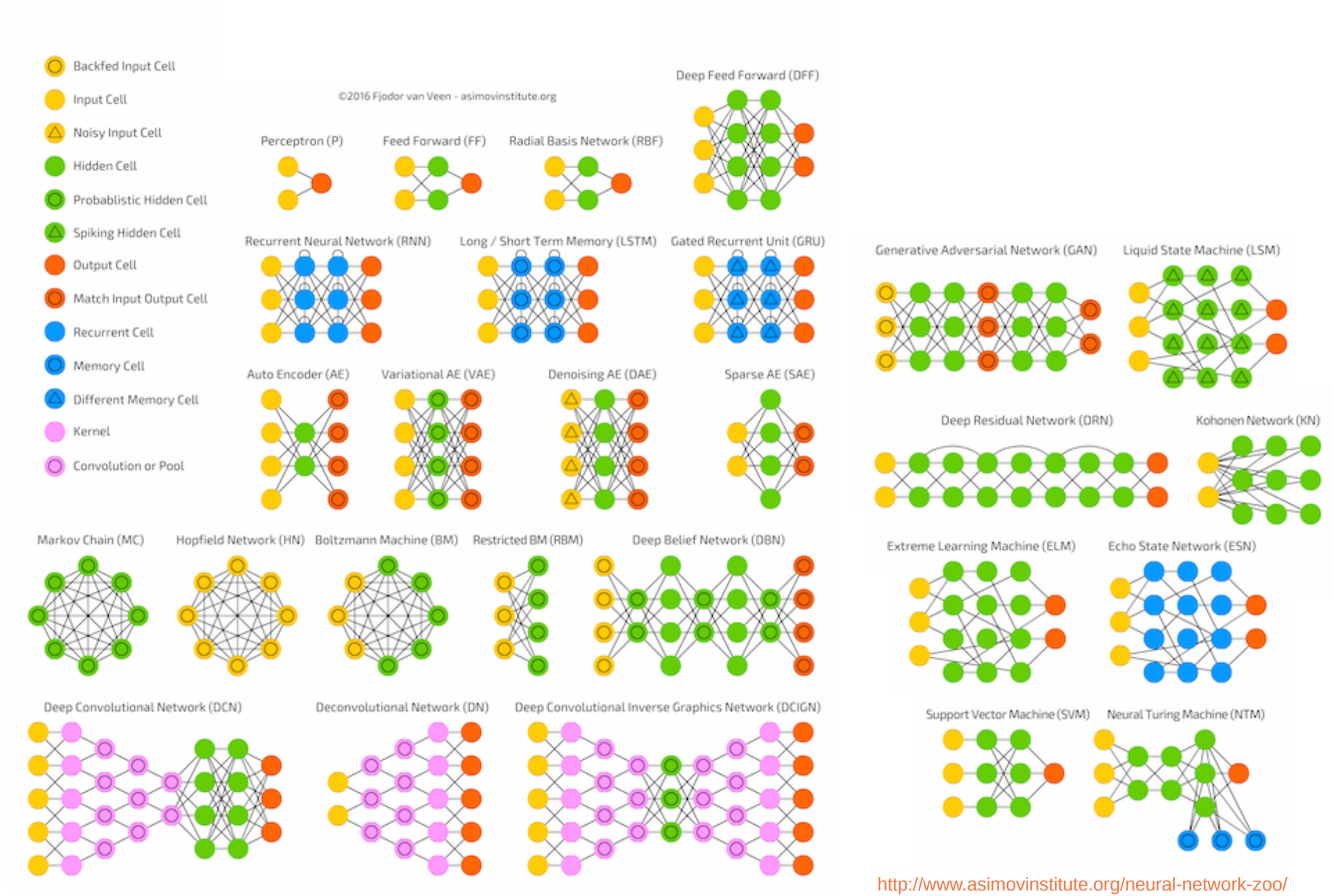
[Ioffe and Szegedy, 2015]

Domain Adaptation



[Mancini et al., CVPR 2018], [Carlucci *et al.*, ICCV2017]

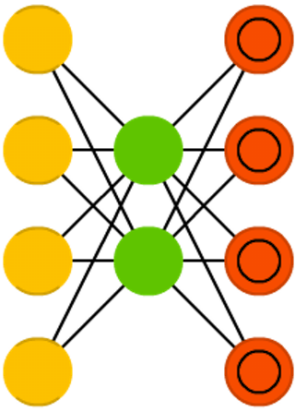
Deep Learning Models



http://www.asimovinstitute.org/neural-network-zoo/

Autoencoders

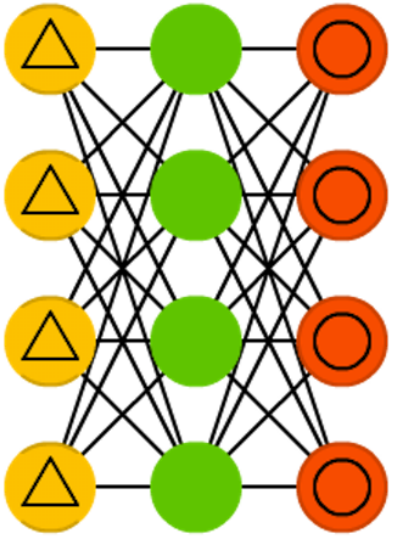
- Unsupervised learning.
- Compress (encode) information automatically.
  - An *encoder* is a deterministic mapping ***f*** that transforms an input vector ***x*** into hidden representation ***y***
  - A *decoder* maps back the hidden representation ***y*** to the reconstructed input ***z*** via ***g***.
- ***Autoencoder:*** compare the reconstructed input ***z*** to the original input ***x*** and try to minimize the reconstruction error.

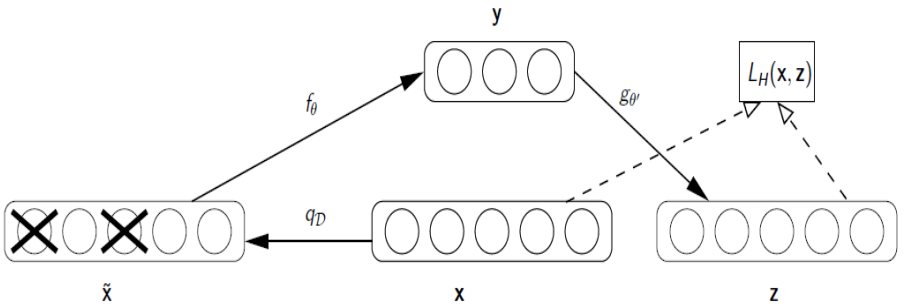


Bourlard, Hervé, and Yves Kamp. “Auto-association by multilayer perceptrons and singular value decomposition.” Biological cybernetics 59.4-5 (1988): 291-294.

Denoising Autoencoders

- Vincent et al. (2010): "*a **good representation** can be obtained **robustly** from a **corrupted input***"
- The higher level representations are relatively stable and robust to input corruption.





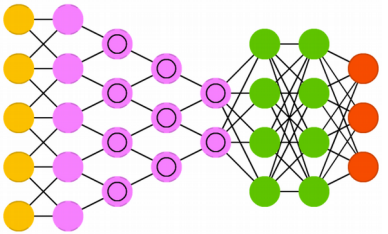
Vincent, Pascal, et al. "Extracting and composing robust features with denoising autoencoders." ICML, 2008.

Structured Data

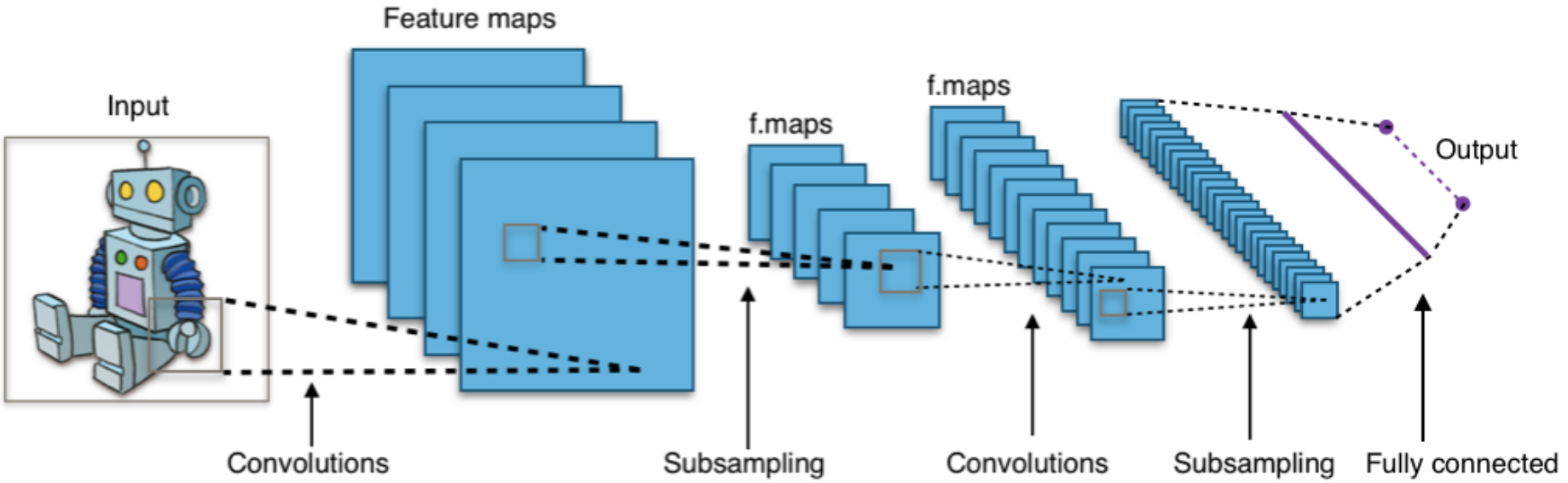
- Some applications naturally deal with an input space which is locally structured – spatial or temporal
- Images, language, etc. vs arbitrary input features
- Deep Learning extremely powerful in this case.

Tomorrow, and tomorrow, and tomorrow; creeps in this petty pace from day to day, until the last syll-able of recorded time. And all our yesterdays have lighted fools the way to dusty

Convolutional Neural Networks

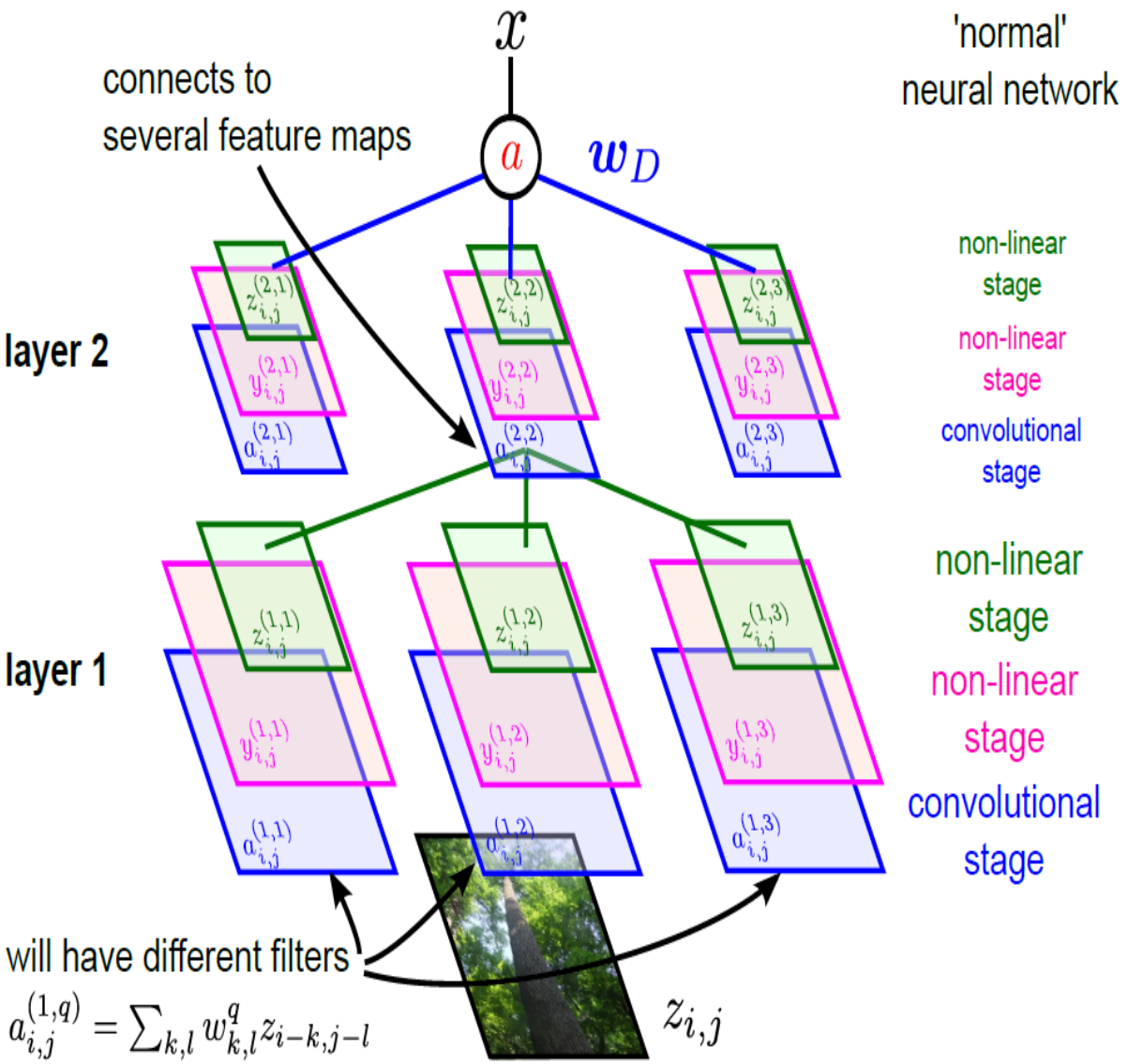


- A multi-layer neural network:
  - With **local connectivity**
  - **Sharing** weight parameters across spatial positions (shift-invariant kernels)



Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11):2278-2324, November 1998

CNN Architecture



$x_{i,j} = \max_{|k|<\tau,|l|<\tau} y_{i-k,j-l}$ pooling stage

mean or subsample also used

$y_{i,j} = f(a_{i,j})$

e.g. $f(a) = [a]_+$

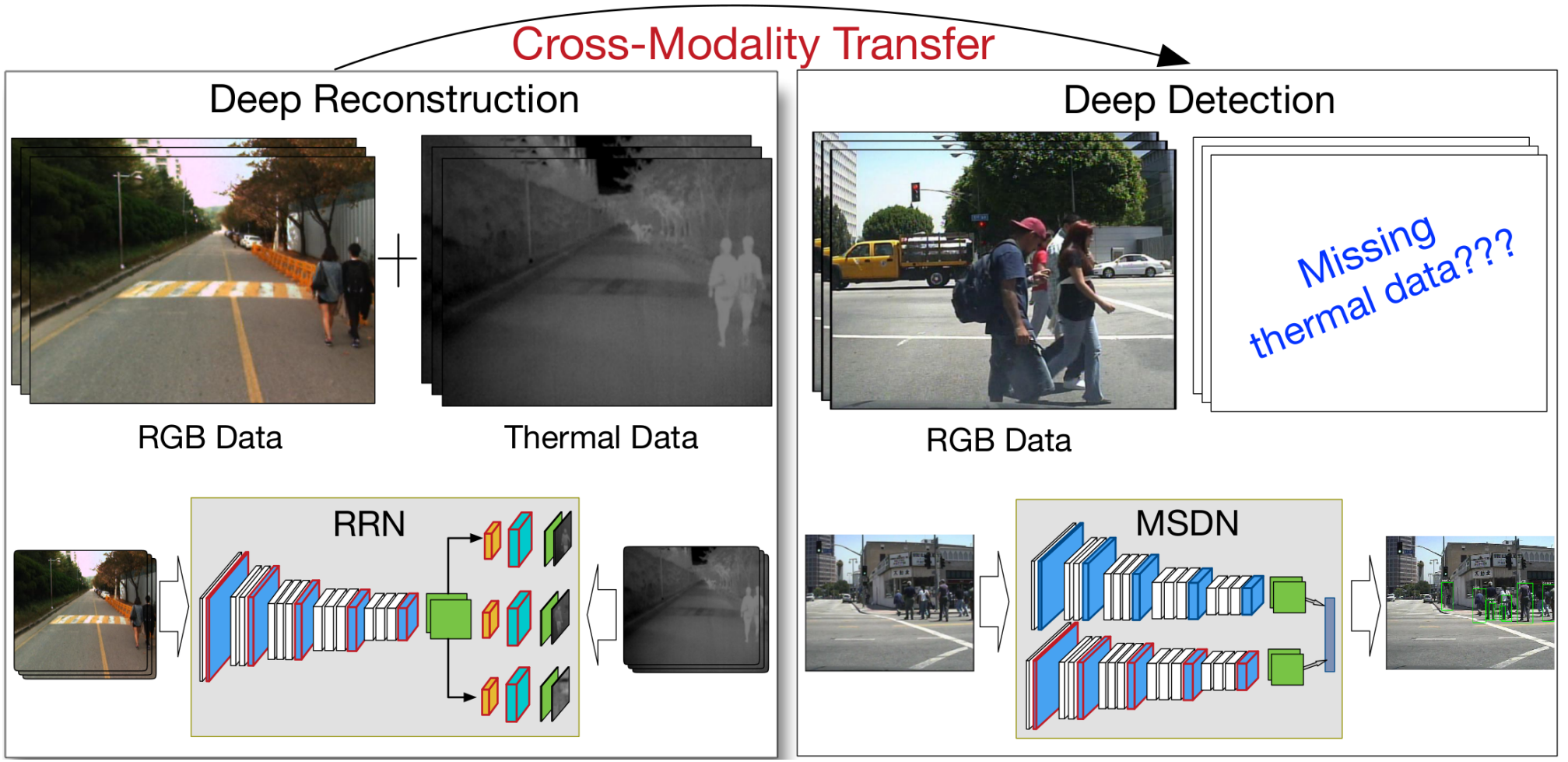
$f(a) = \text{sigmoid}(a)$

non-linear stage

$a_{i,j} = \sum_{k,l} w_{k,l} z_{i-k,j-l}$ convolutional stage

only parameters

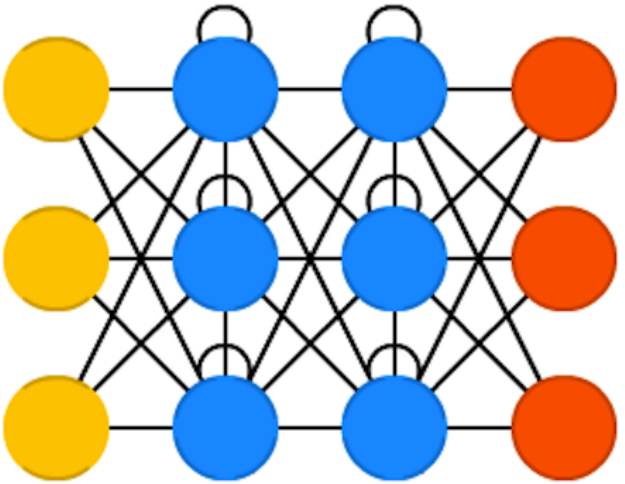
Pedestrian Detection



[Xu et al., CVPR 2017]

Recurrent Neural Networks

- Standard Neural Networks (and also CNN):
  - Accept a fixed-size vector/matrix as input (e.g., an image) and produce a fixed-size vector as output (e.g., probabilities of different classes).
  - Use a fixed amount of computational steps (e.g. the number of layers in the model).
- Recurrent Neural Networks are unique as they permit to operate on sequences of vectors.
- Sequences in the input, the output, or both.



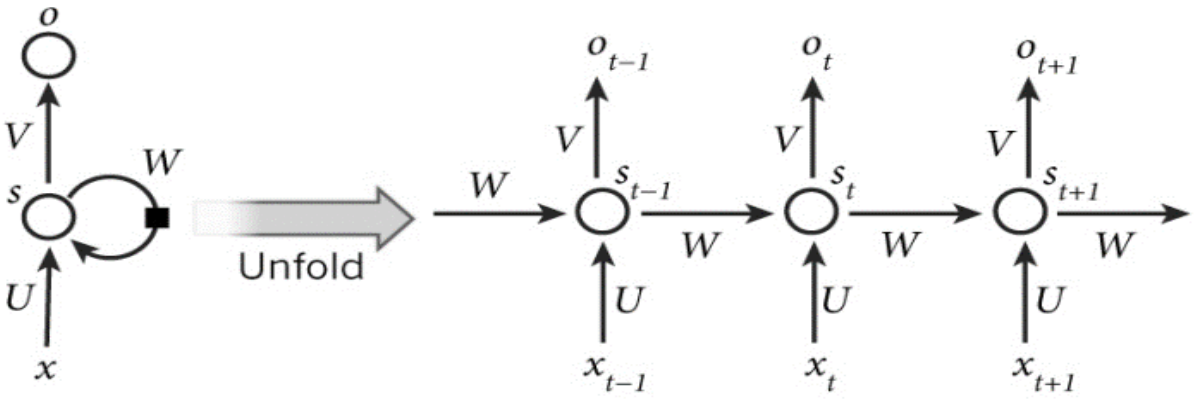
Elman, Jeffrey L. “Finding structure in time.” Cognitive science 14.2 (1990): 179-211

Recurrent Neural Networks

- An unrolled RNN (in time) can be considered as a deep neural network with indefinitely many layers:

$$s_t = f(Ux_t + Ws_{t-1})$$

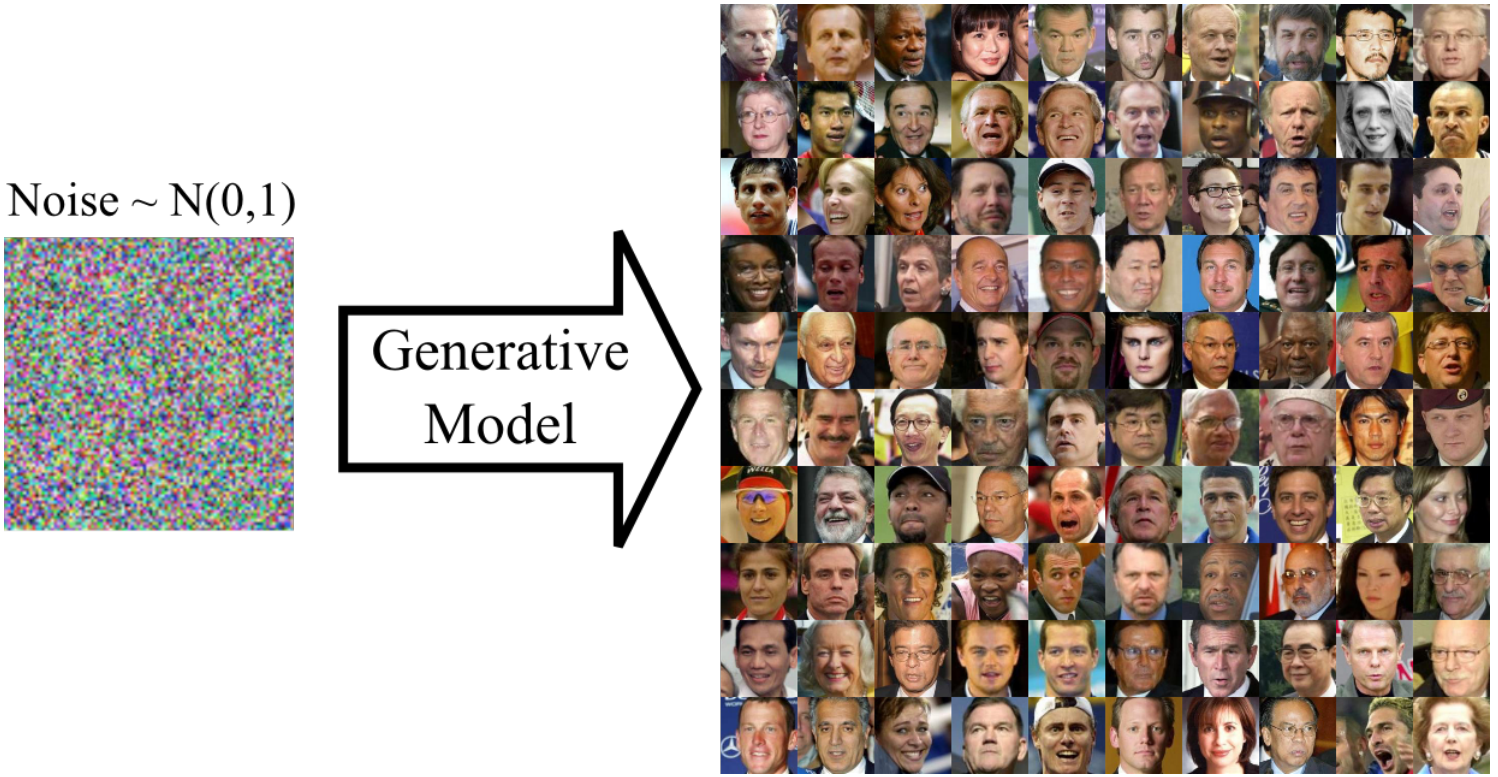
$$y = g(Vs_t)$$



- The parameters to be learned (U, V, W) are ***shared*** by all time steps in the network. The gradient at each output depends not only on the calculations of the current time step but also of the previous time steps.

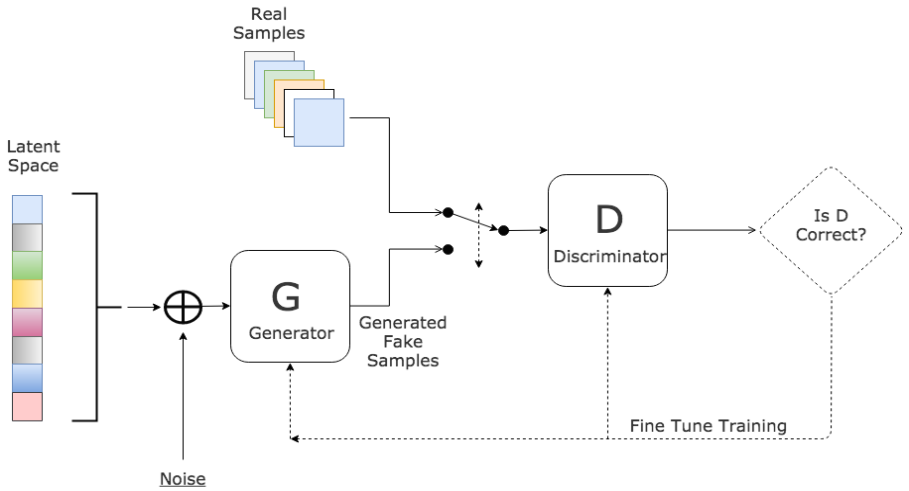
Deep Generative Models

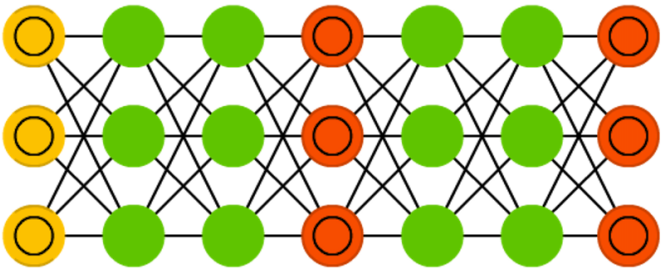
- Lots of research on generative models to create probabilistic models of training data with ability to generate new images, sentences, etc.



Generative Adversarial Networks (GANs)

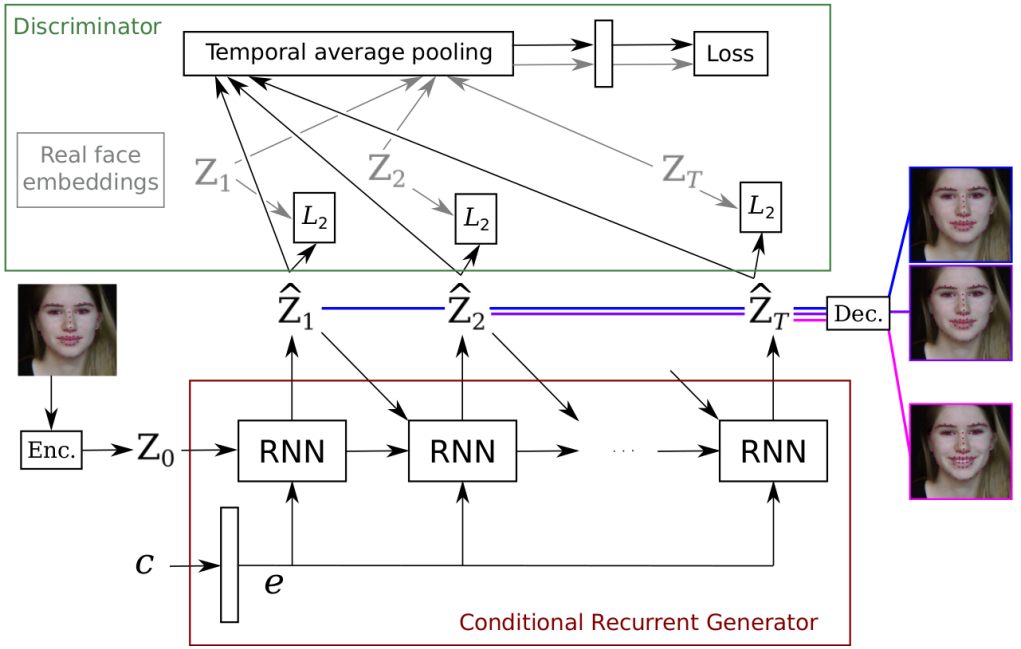
- Generator net produces samples *x* close to training samples
- Discriminator net (adversary) must differentiate between samples from the generative net and the training set
- Use error feedback to improve task of both nets, until discriminator can no longer distinguish
- Discriminator net is discarded at test time.

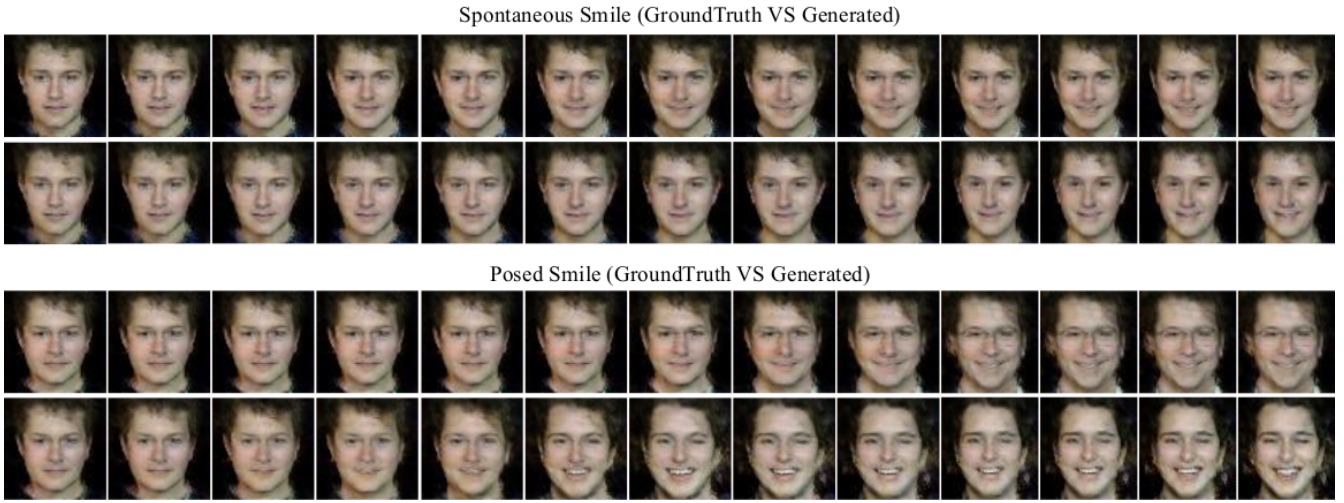




Goodfellow, Ian, et al. “Generative adversarial nets.” Advances in Neural Information Processing Systems. 2014.

Video Generation

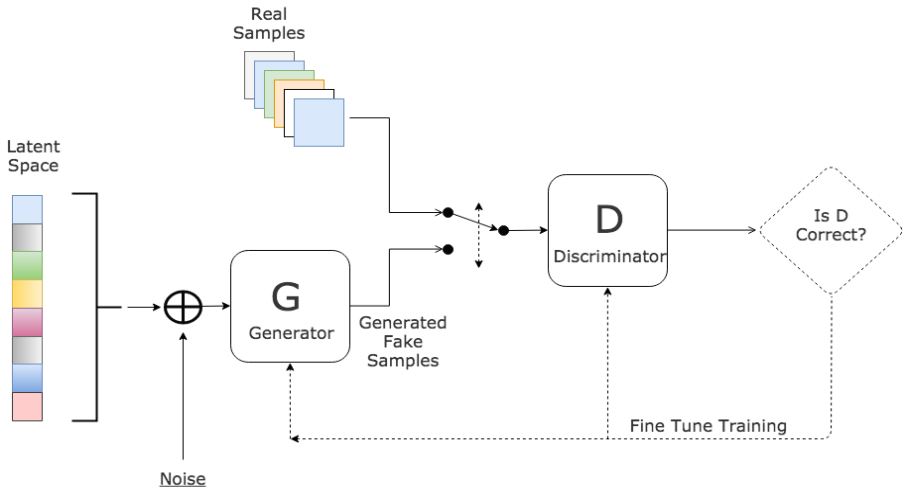




[Wang et al., CVPR 2018]

Video Generation

- Generator net produces samples *x* close to training samples
- Discriminator net (adversary) must differentiate between samples from the generative net and the training set
- Use error feedback to improve task of both nets, until discriminator can no longer distinguish
- Discriminator net is discarded at test time.

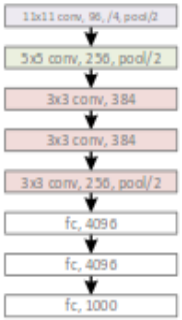


Goodfellow, Ian, et al. “Generative adversarial nets.” Advances in Neural Information Processing Systems. 2014.

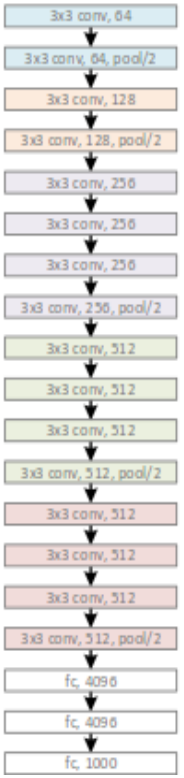
Open Issues: NN size

Scale: larger and larger nets...

AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)

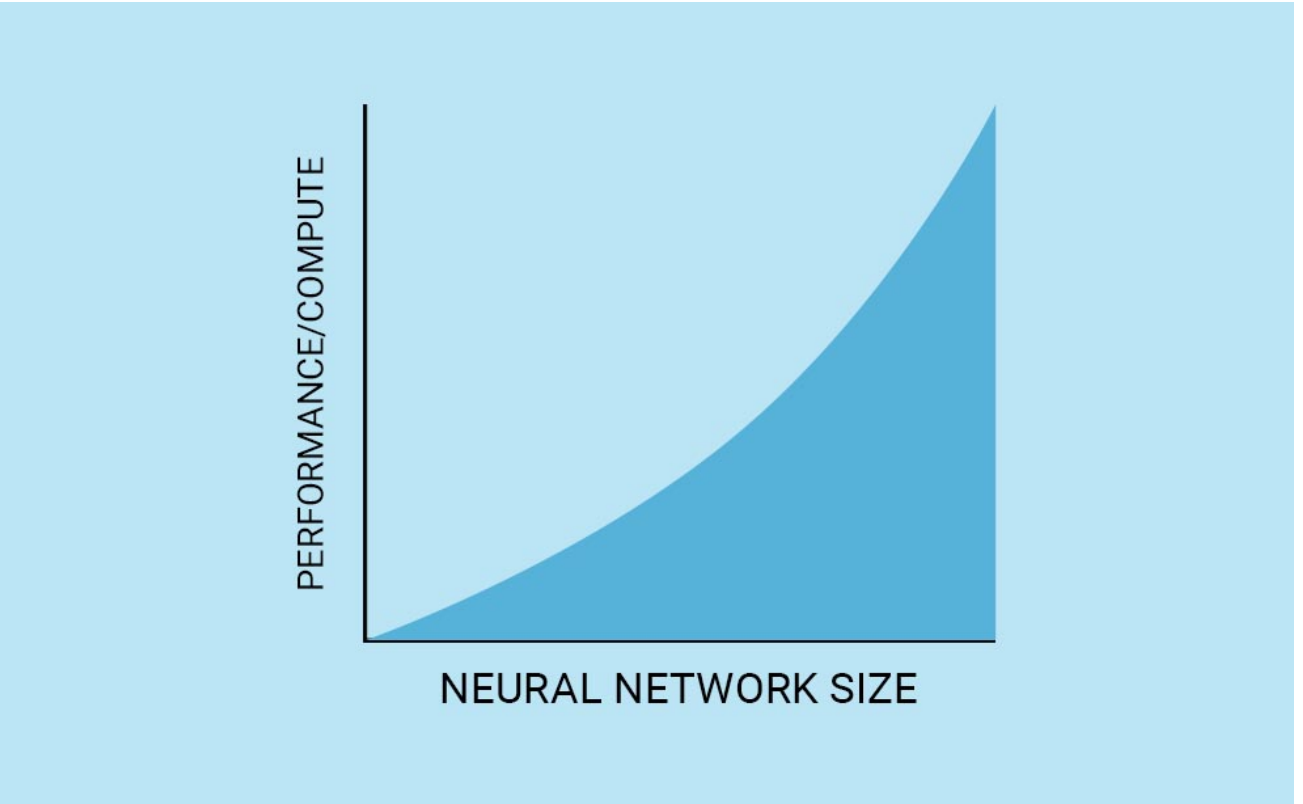


GoogleNet, 22 layers
(ILSVRC 2014)

ResNet, 152 layers
(ILSVRC 2015)

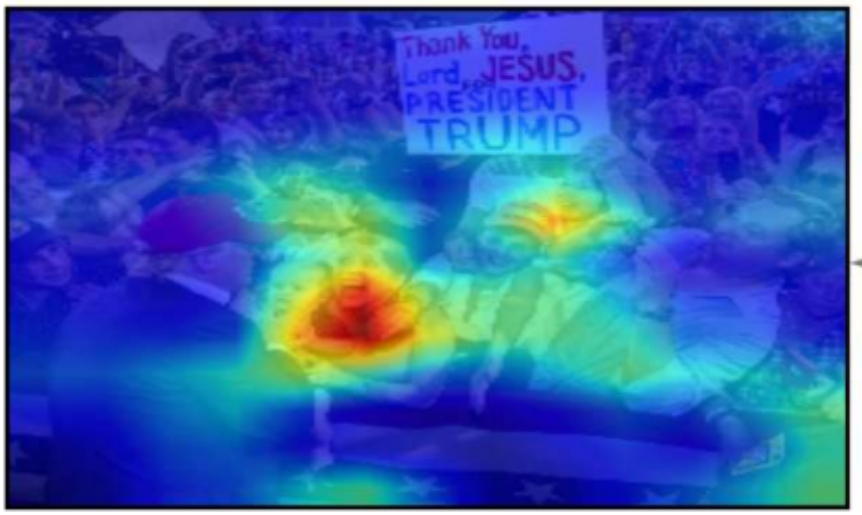
Open Issues: NN size

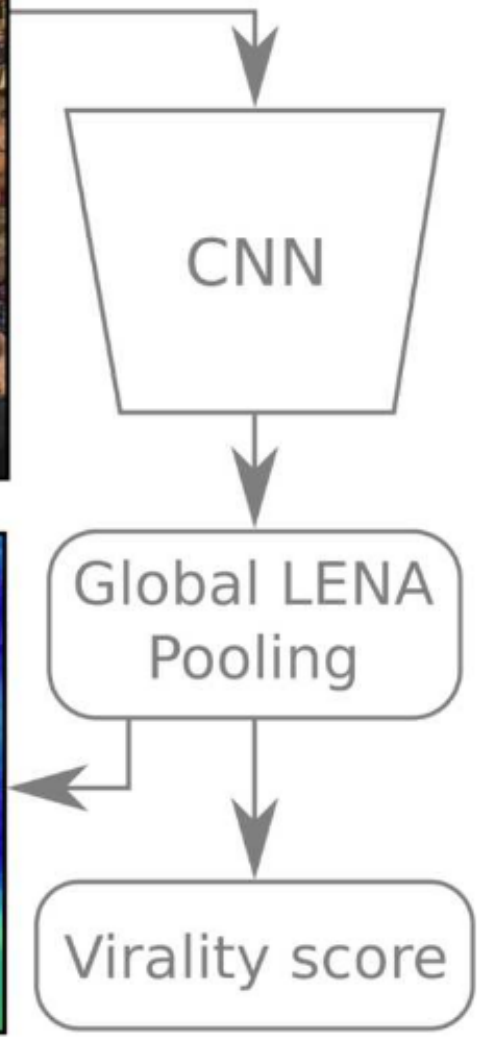
Scale: how to stop this???



Open Issues: Unsupervised Learning

Open Issues: XAI





[Alameda-Pineda, CVPR2017]

Thanks for your attention!