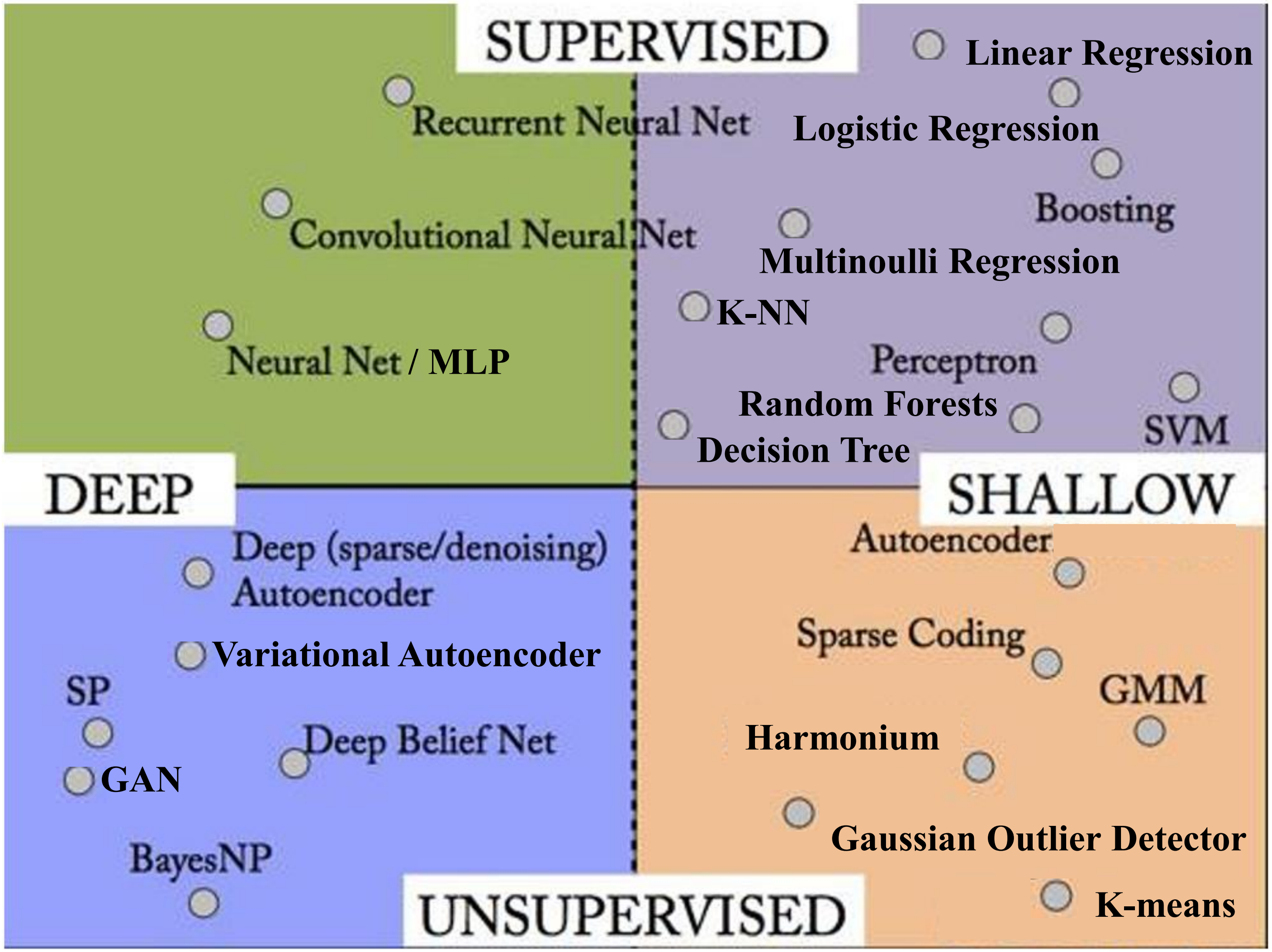
Into the Black Box:
On Artificial Neural Networks

Alexander G. Ororbia II

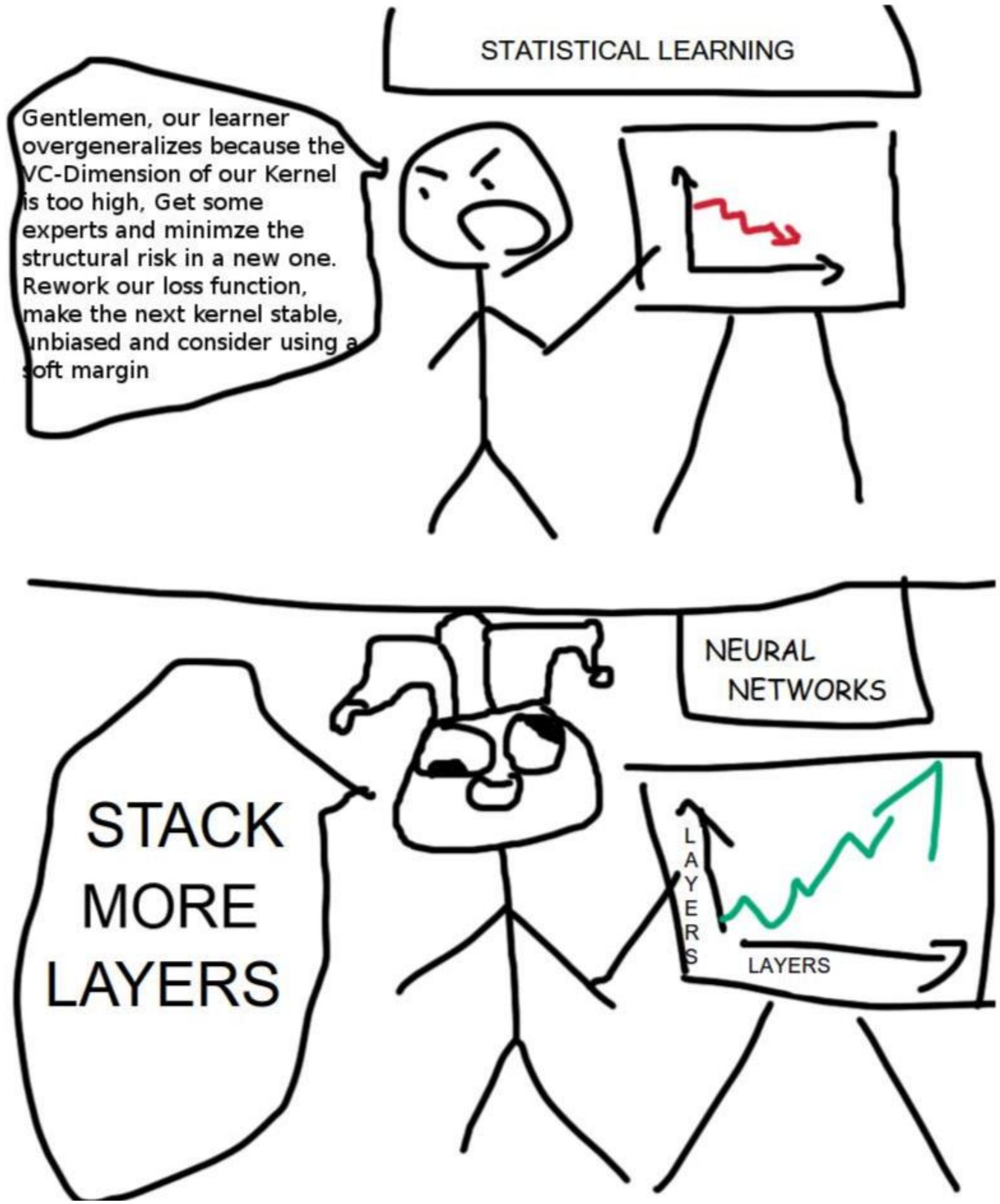
Introduction to Machine Learning

CSCI-335

4/15/2026


SUPERVISED
Linear Regression
Recurrent Neural Net
Logistic Regression
Boosting
Convolutional Neural Net
Multinoulli Regression
K-NN
Neural Net / MLP
Perceptron
Random Forests
SVM
Decision Tree
DEEP
SHALLOW
Deep (sparse/denoising) Autoencoder
Autoencoder
Variational Autoencoder
Sparse Coding
SP
GMM
Harmonium
Deep Belief Net
GAN
Gaussian Outlier Detector
BayesNP
UNSUPERVISED
K-means

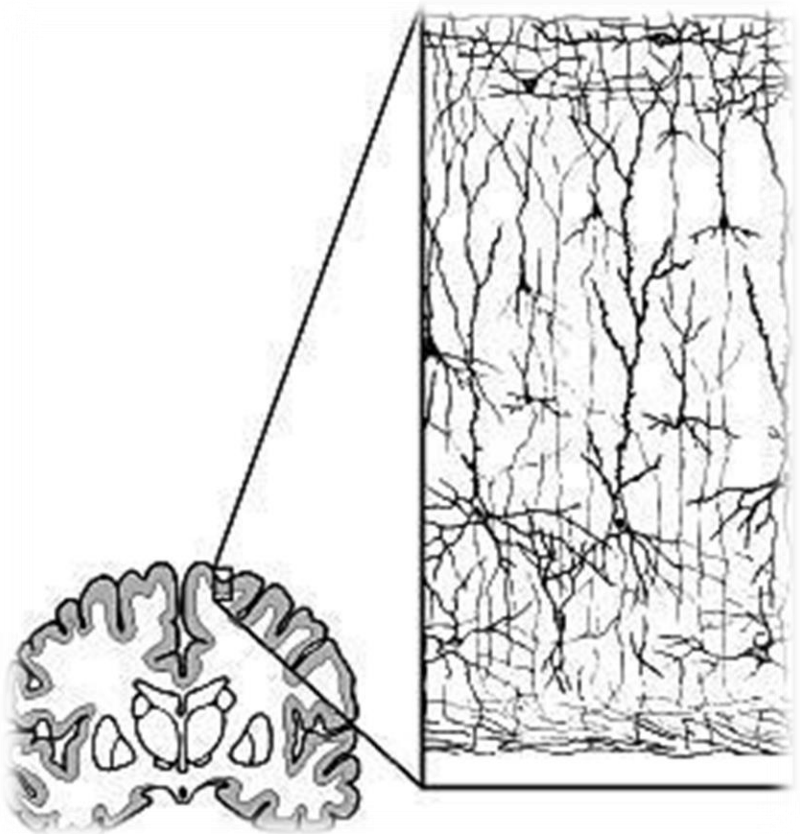
We still have a lot to learn….



https://www.reddit.com/r/machinegoofingoff/

Artificial Neural Networks (ANNs): Neurobiological Motivations

- Human ***brain*** = a good candidate inference engine and learning algorithm
  - Evidence of layered architectures in neuro-scientific research, i.e., cortical structures
- Early success of specialized yet deep architectures
  - Convolutional Networks, NeoCognitron



http://cs.brown.edu/~tld/projects/cortex/

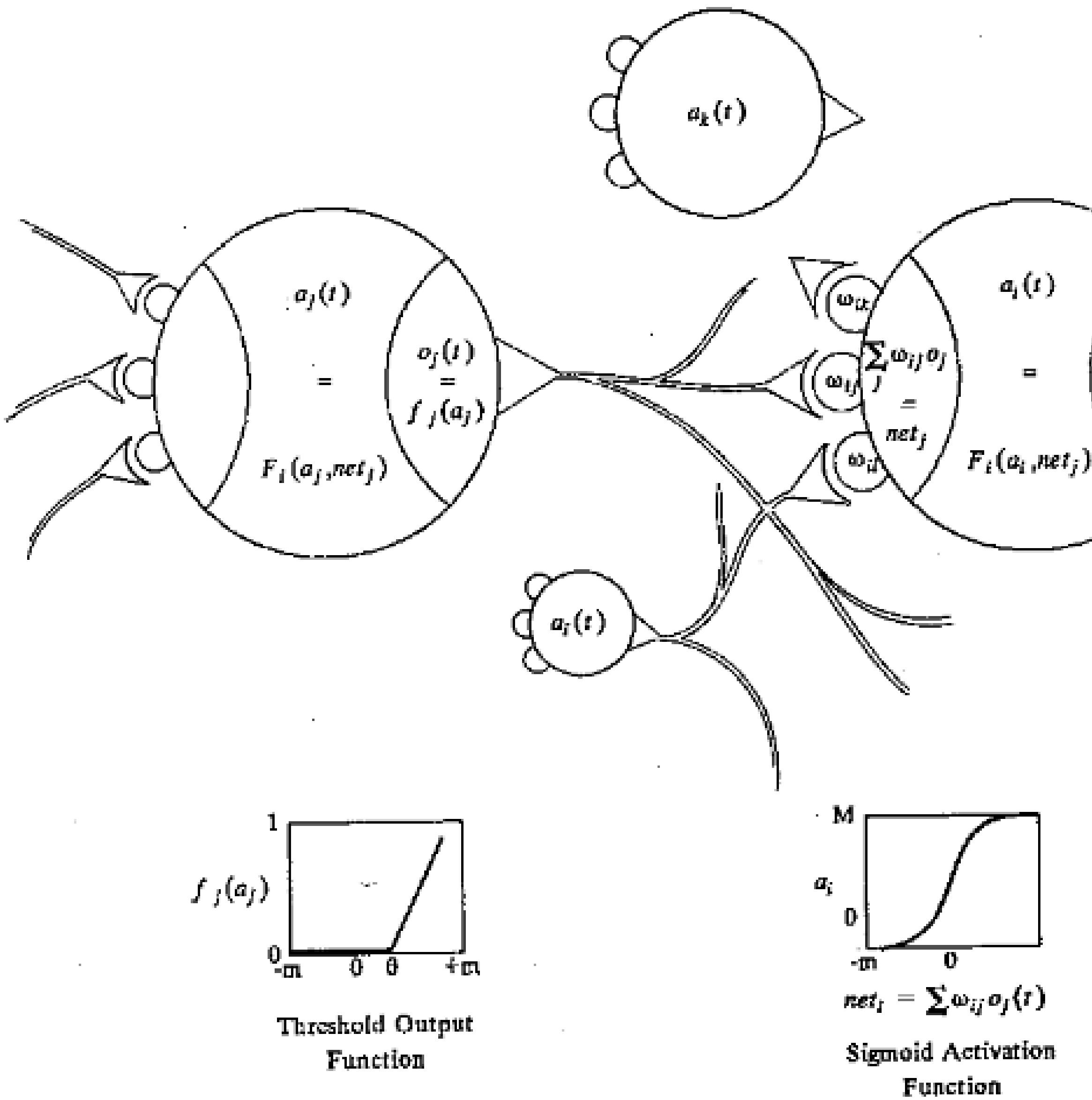
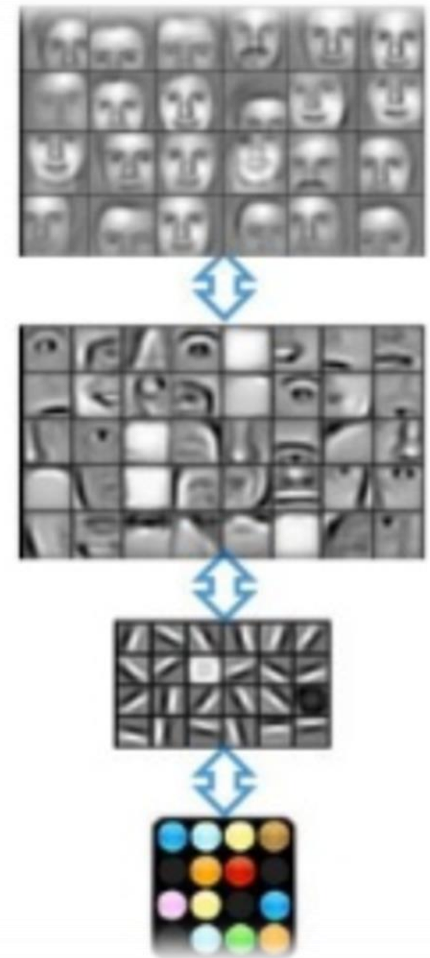


FIGURE 1. The basic components of a parallel distributed processing system.

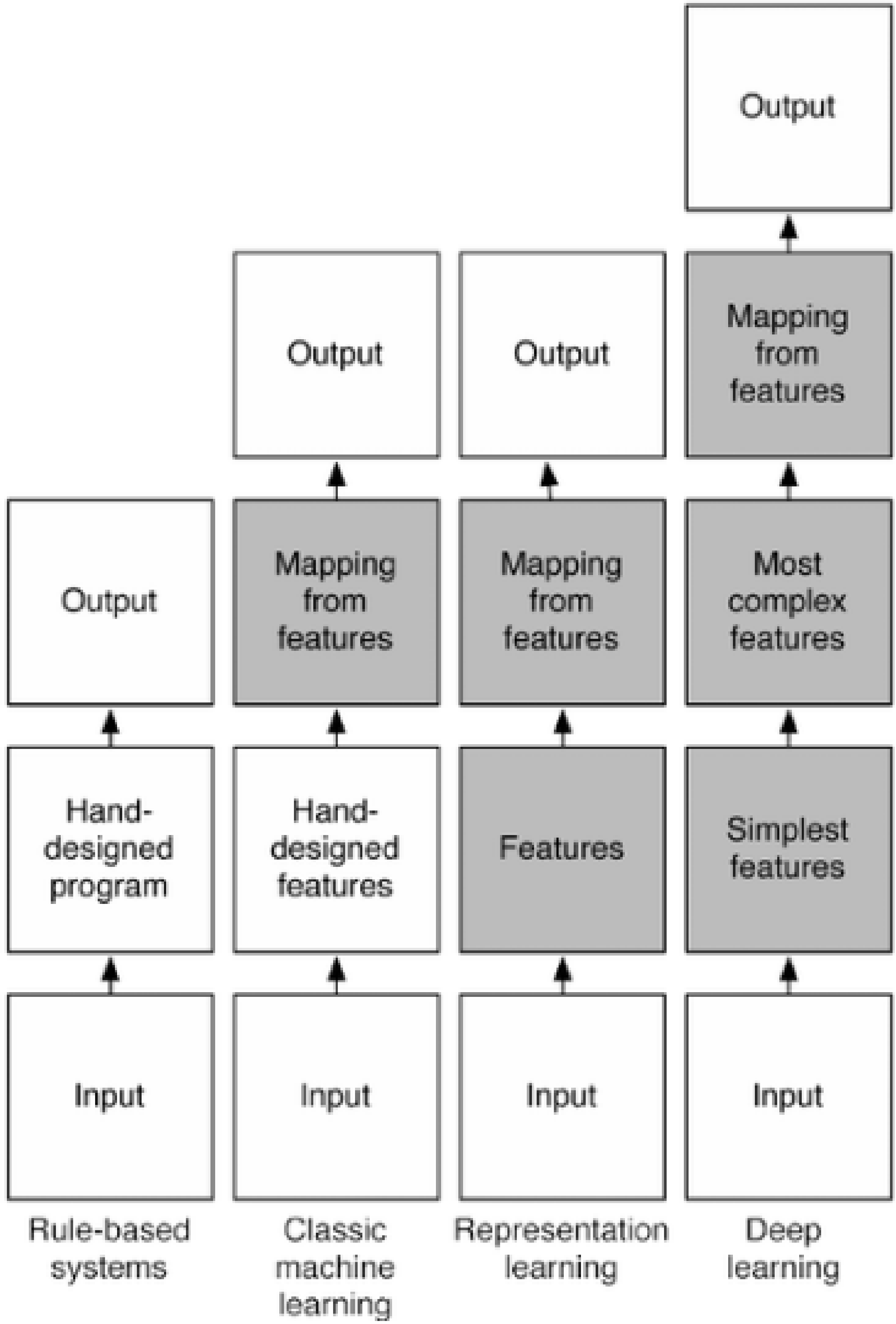
An organic take on a neural system (Rummelhart, Hinton, & McClelland, 1986).

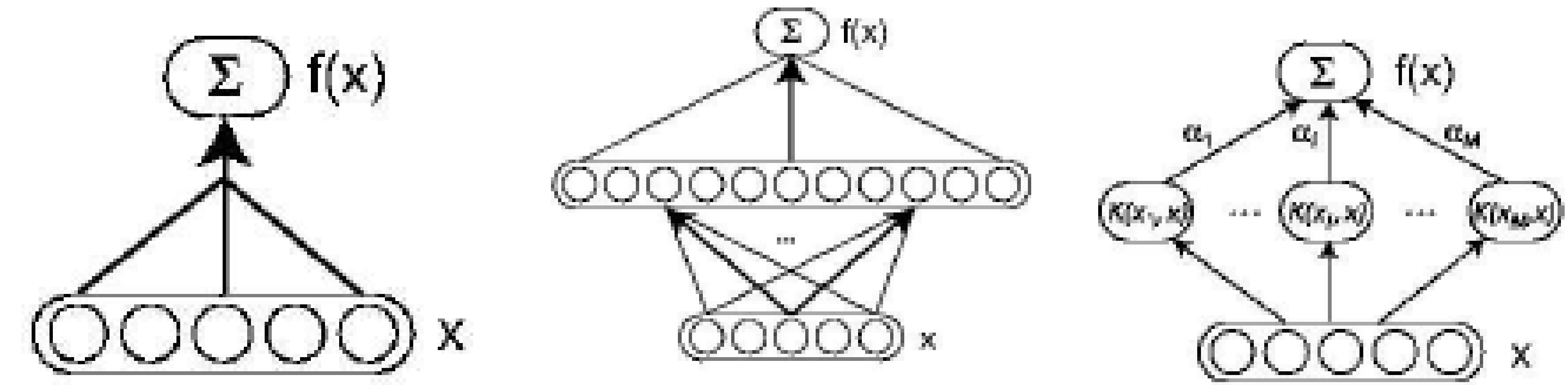
Why? Feature Abstraction

- Raw features, e.g., pixel values of image, viewed as "low-level" data representation
  - Can be complex & high-dimensional
  - Observed variables: "nature" sensory samples, observed/recorded
- **Abstract representations** = layers of feature detectors
  - Latent /unobserved variables that describe observed variables
    - Capture key aspects of underlying stochastic process
    - Many concepts can be represented as (strict) hierarchies (taxonomy of species) → goal of model is to "learn" a plausible, structured unknown hierarchy
  - Idea: extracting "structure" from "unstructured"/messy data
- **Automatic feature engineering**



http://www.slideshare.net/roelofp/2014-1021-sicsdlnlpg


Output
Output
Output
Mapping from features
Output
Mapping from features
Mapping from features
Most complex features
Hand-designed program
Hand-designed features
Features
Simplest features
Input
Input
Input
Input
Rule-based systems
Classic machine learning
Representation learning
Deep learning

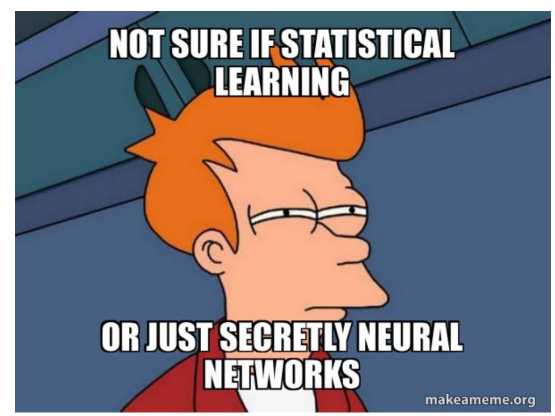


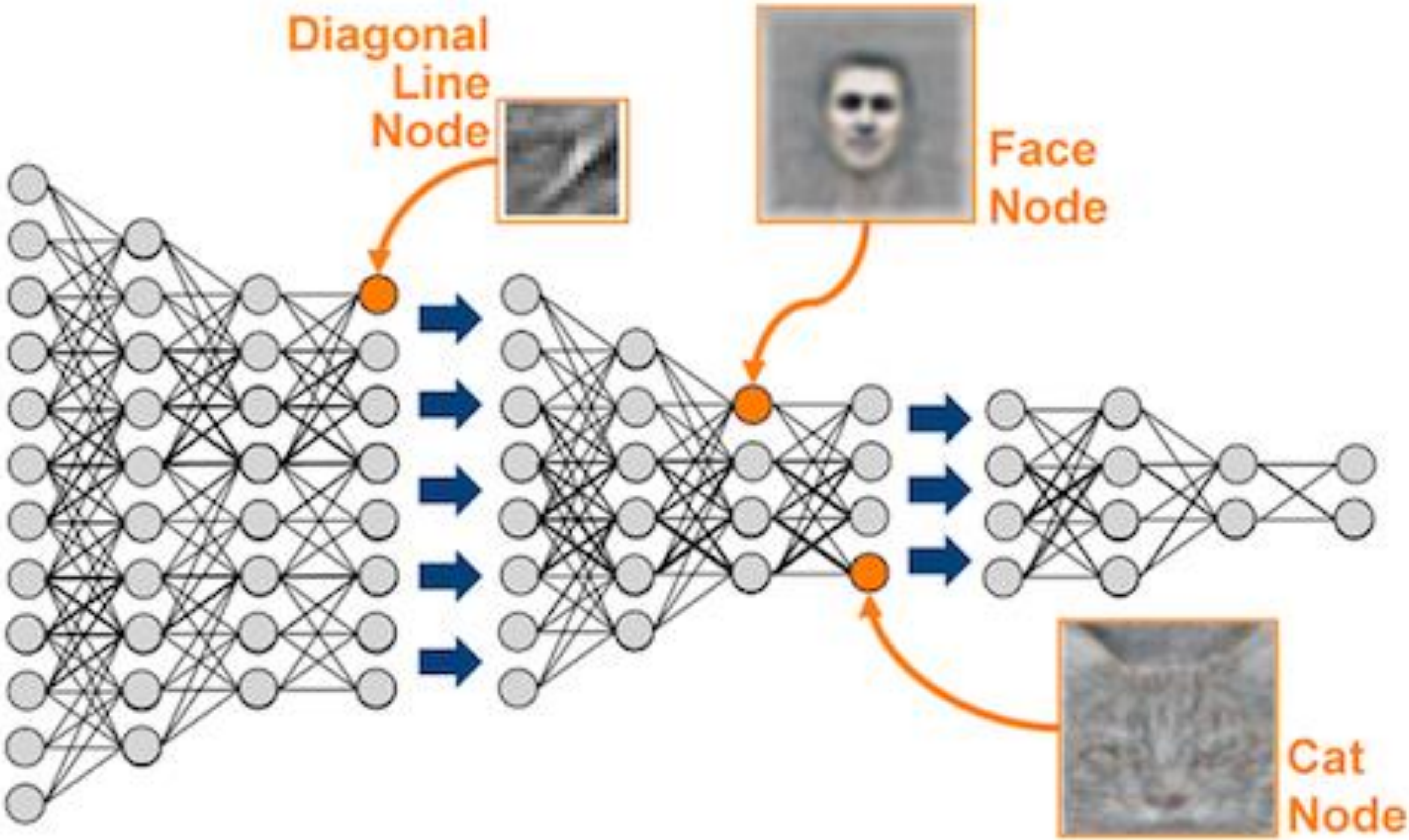
(a) Linear model architecture

(b) Single layer neural network architecture

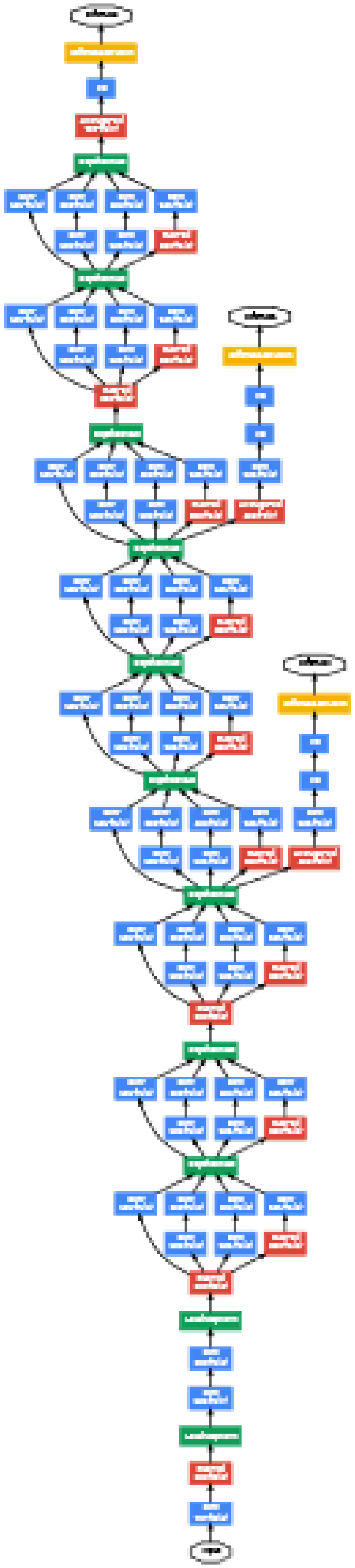
(c) Kernel SVM architecture

Most of machine learning can be viewed as a type of ANN…if you squint hard enough…

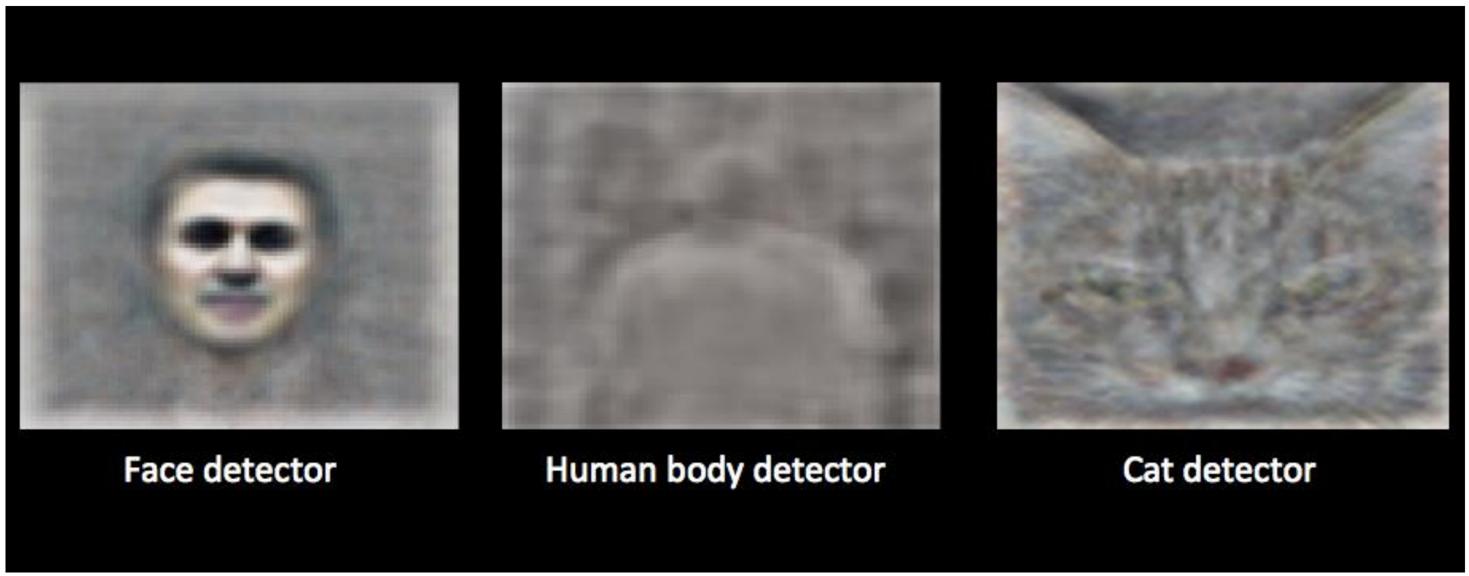


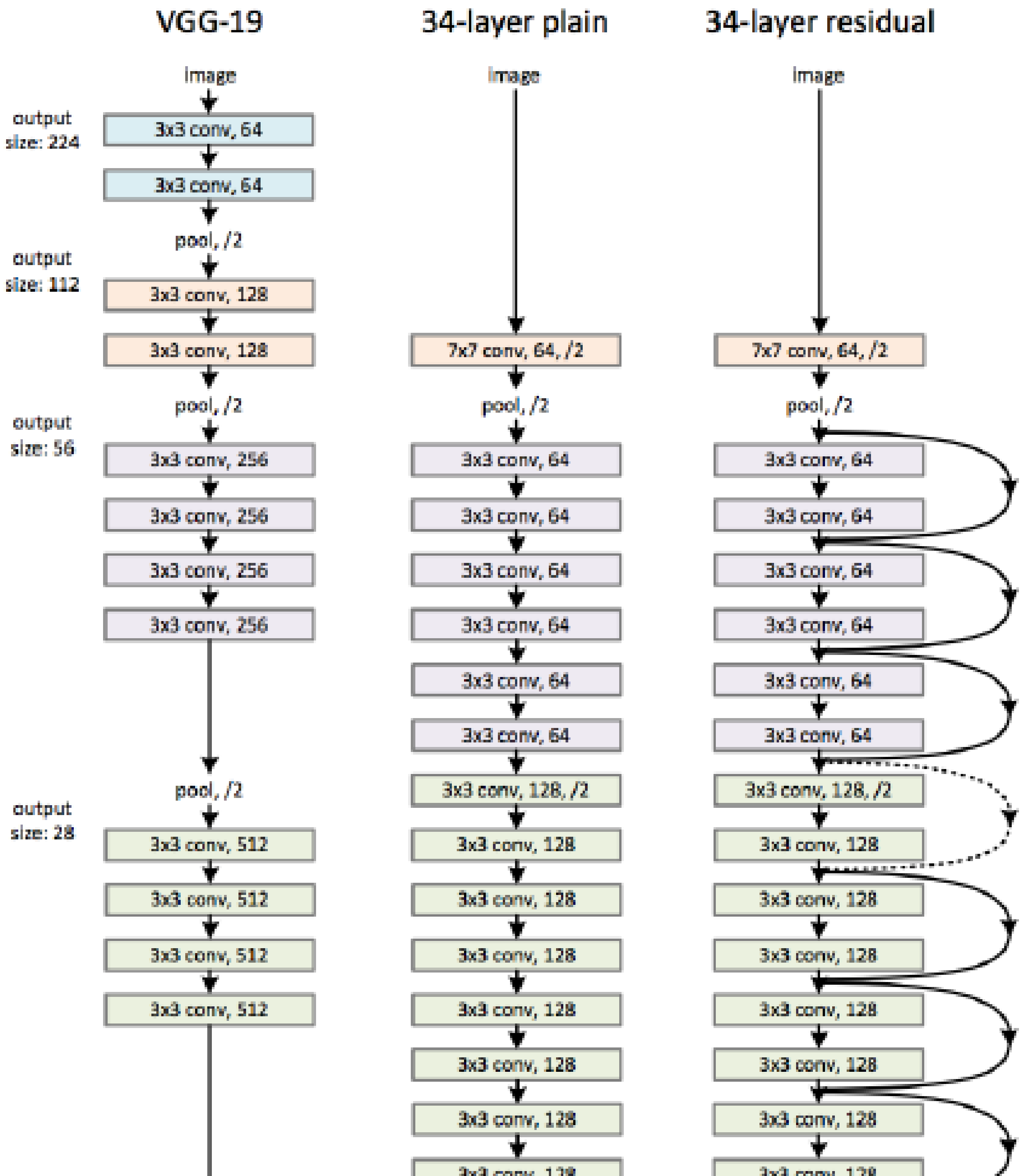


Diagonal Line Node
Face Node
Cat Node

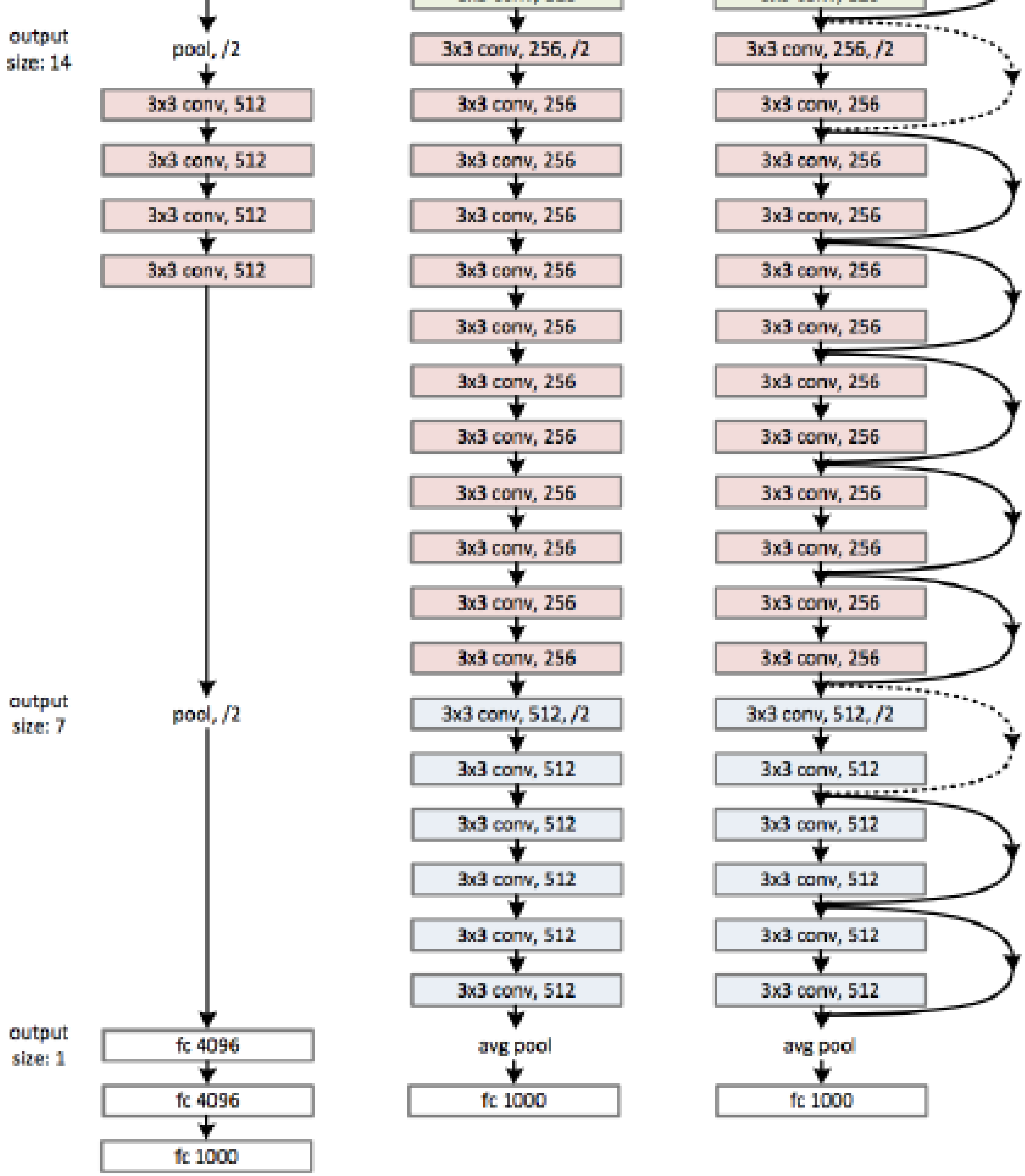


Deep cat detector!





VGG-19
34-layer plain
34-layer residual
image
output size: 224
3x3 conv, 64
3x3 conv, 64
pool, /2
output size: 112
3x3 conv, 128
3x3 conv, 128
pool, /2
output size: 56
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool, /2
output size: 28
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128



output size: 14
pool, /2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
output size: 7
pool, /2
output size: 1
fc 4096
fc 4096
fc 1000
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
avg pool
fc 1000

Background

A Recipe for Machine Learning

1. Given training data:

$$\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$$

2. Choose each of these:
   - Decision function

$$\hat{\boldsymbol{y}} = f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$$

   - Loss function

$$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}_i) \in \mathbb{R}$$

3. Define goal:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^N \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i)$$

4. Train with SGD:
(take small steps opposite the gradient)

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta_t \nabla \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i)$$

Gradients

Backpropagation can compute this gradient!

And it's a **special case of a more general algorithm** called reverse-mode automatic differentiation that can compute the gradient of any differentiable function efficiently!

Background

A Recipe for

1. Given training dat

$\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$

2. Choose each of t

– Decision functio

$\hat{y} = f_{\boldsymbol{\theta}}(\boldsymbol{x}_i)$

– Loss function

$\ell(\hat{\boldsymbol{y}}, \boldsymbol{y}_i) \in \mathbb{R}$

opposite the gradient)

$\boldsymbol{\theta}^{(t} \ldots {}^{(t)} - \eta_t \nabla \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}_i), \boldsymbol{y}_i)$

Artificial Neural Networks: Dynamics

Understanding the multilayer perceptron (MLP

QUESTIONS?