SAND2018-8459PE

Deep Learning MLDL 2018 Tutorial

Tim Draelos

July 19, 2018

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Deep Learning

- Definitions
- Concepts
- Architectures/Algorithms
- Applications
- Development Stages
- Software
- Hardware
- Data
- Demonstration
- Current Research Topic

Machine Learning Fundamentals

- What is Machine Learning?
  - Machine learning is the process of "learning" a function mapping inputs to outputs

Input → Math → Output

Features

- Time of Day
- Time of Year
- Yesterday's Weather

Model

- Neural Network
- Support Vector Machine
- Decision Tree

Label or Target

- Cold
- Rainy

  - Training Phase – learning from examples of Input/Output pairs
  - Test/Evaluation/Use Phase

Machine Learning

Input → Math → Output

Features | Model | Label or Target

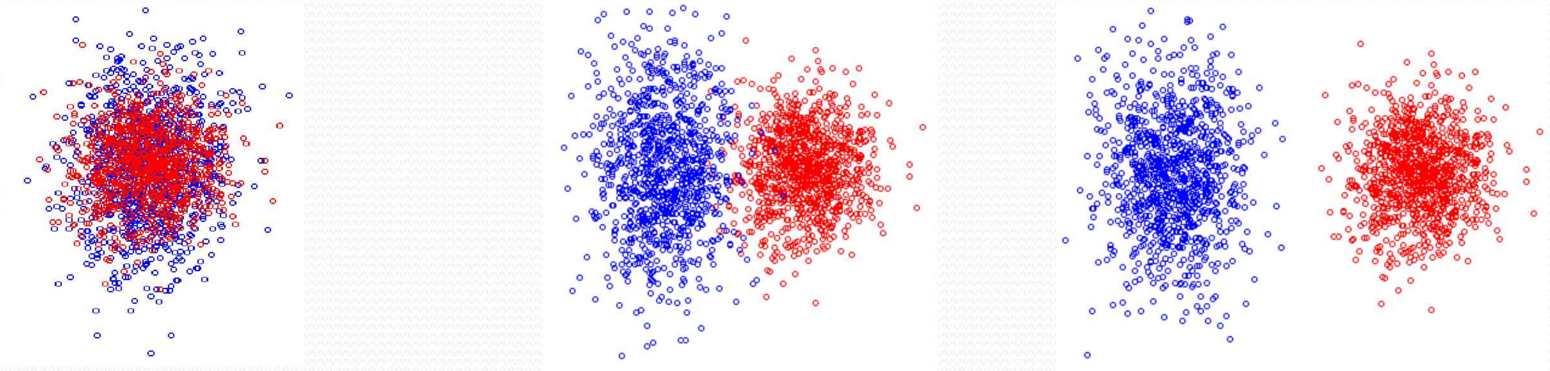
Deep Learning

Input → Math → Output

Data | Artificial Neural Network | Label or Target

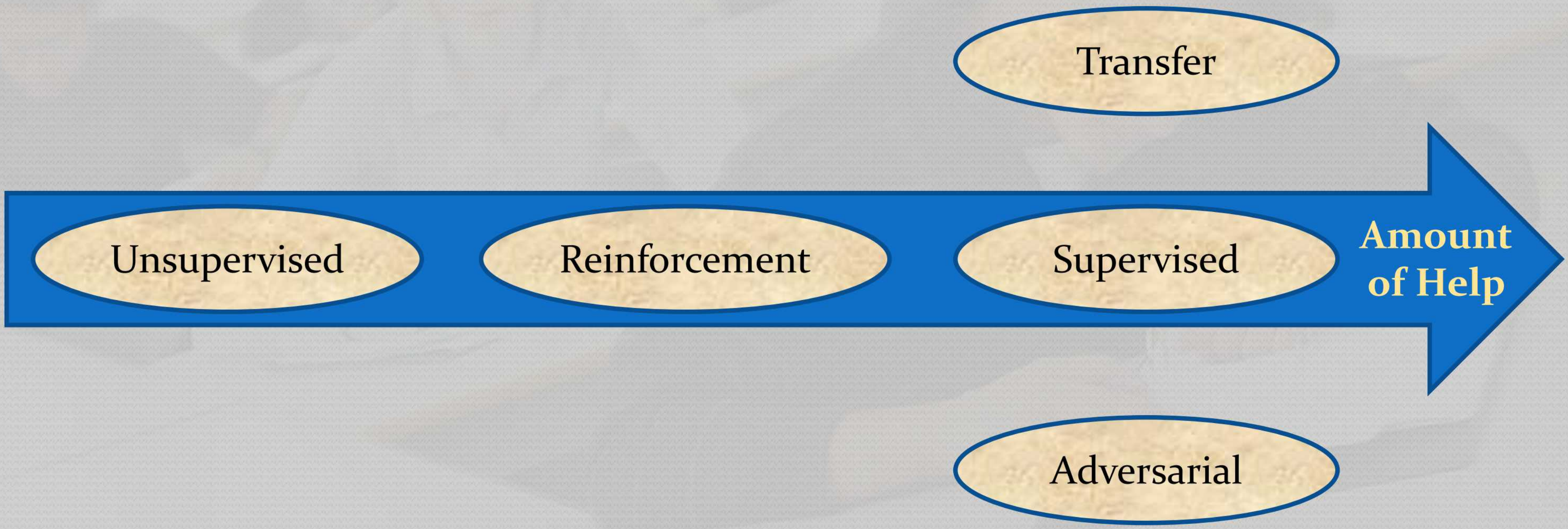
Deep Learning

- **Representation Learning** to Discover/Detect Features
  - Hierarchical Feature Selection / Feature Extraction
    - Features = Explanatory factors, attributes of data samples
  - Good representation (feature set) captures the *a posteriori* distribution of underlying explanatory factors (e.g., Classes) of an observed environment.
    - P(Class | Data) ➔ P(Class | Features)
    - Each data sample is a biased representation of its class



Types of Learning

- **Supervised** – Teacher has right answers
- **Unsupervised** – No teacher, no answers
- **Reinforcement** – Environment teaches with rewards
- **Transfer** – Use previous learning on a new problem
- **Adversarial** – Teacher can't be trusted



Supervised Learning

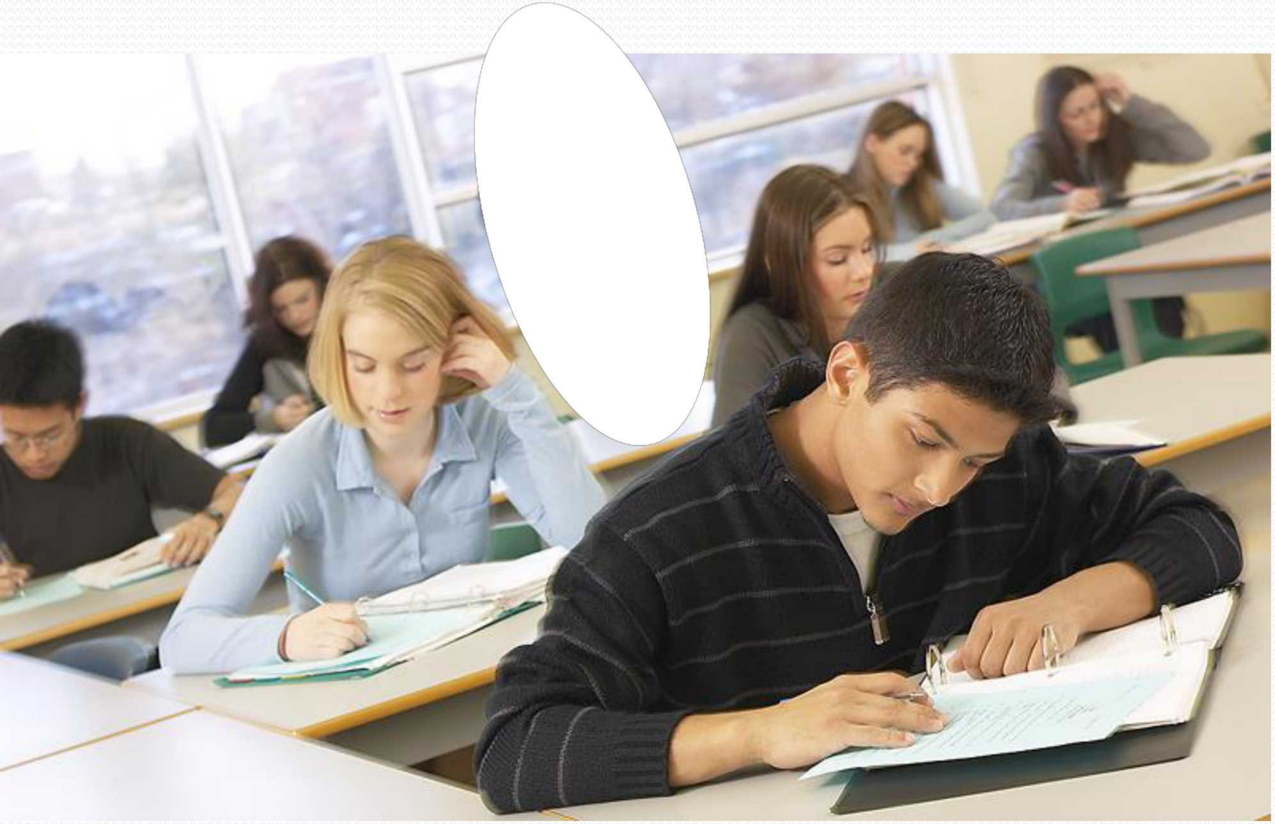
- Teacher has the right answers to each question

- Answers to Questions
  1. C
  2. B
  3. C
  4. A
  5. D
  6. E
  7. B

  …

Unsupervised Learning

- No teacher, no answers, no grade



- 15 Math questions
  - 10 multiple choice
- 20 English questions
  - 7 multiple choice
- 12 History questions

…

Reinforcement Learning

- Environment teaches with rewards
  - No answers to each question, just a grade

- Student 1 – 49%
- Student 2 – 63%
- Student 3 – 87%
- Student 4 – 92%

  …

Transfer Learning

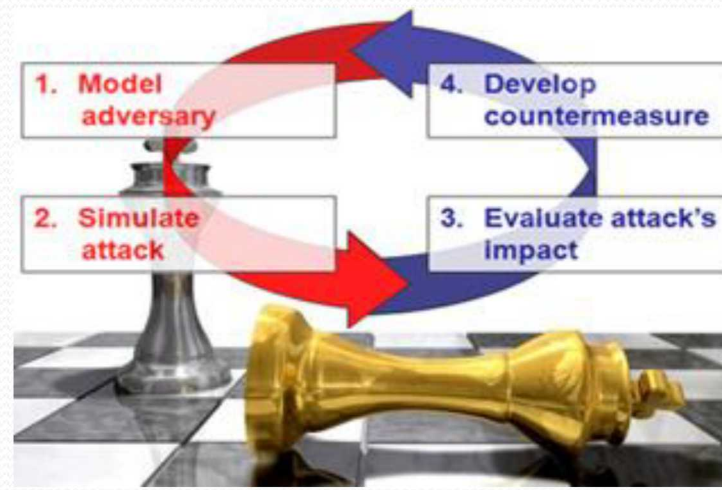
- Use previous learning on new classes
  - Use learning in Math class on Physics Test

Math Class

Physics Test

Adversarial Learning

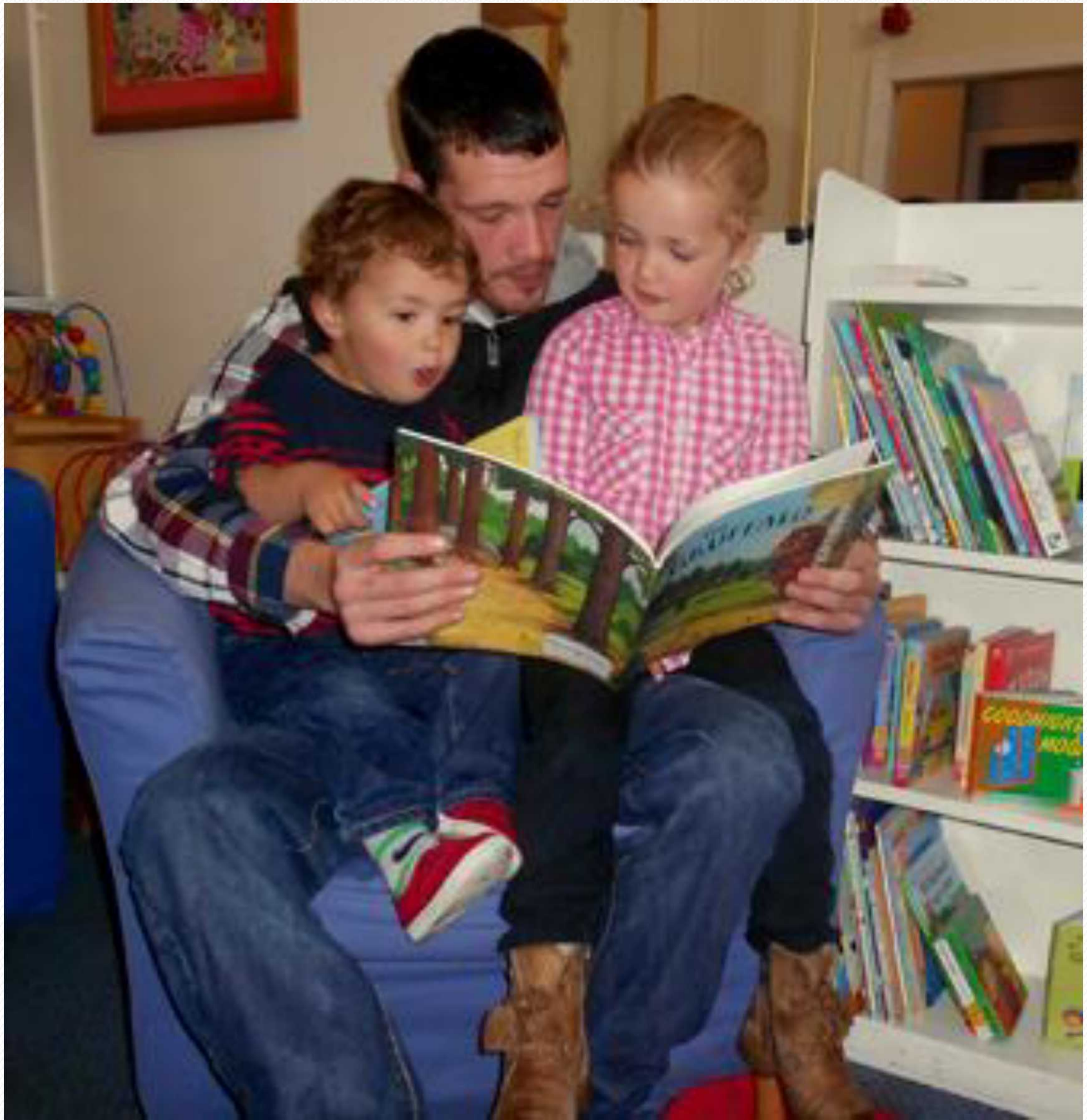
- Teacher can't be trusted



Unsupervised Learning

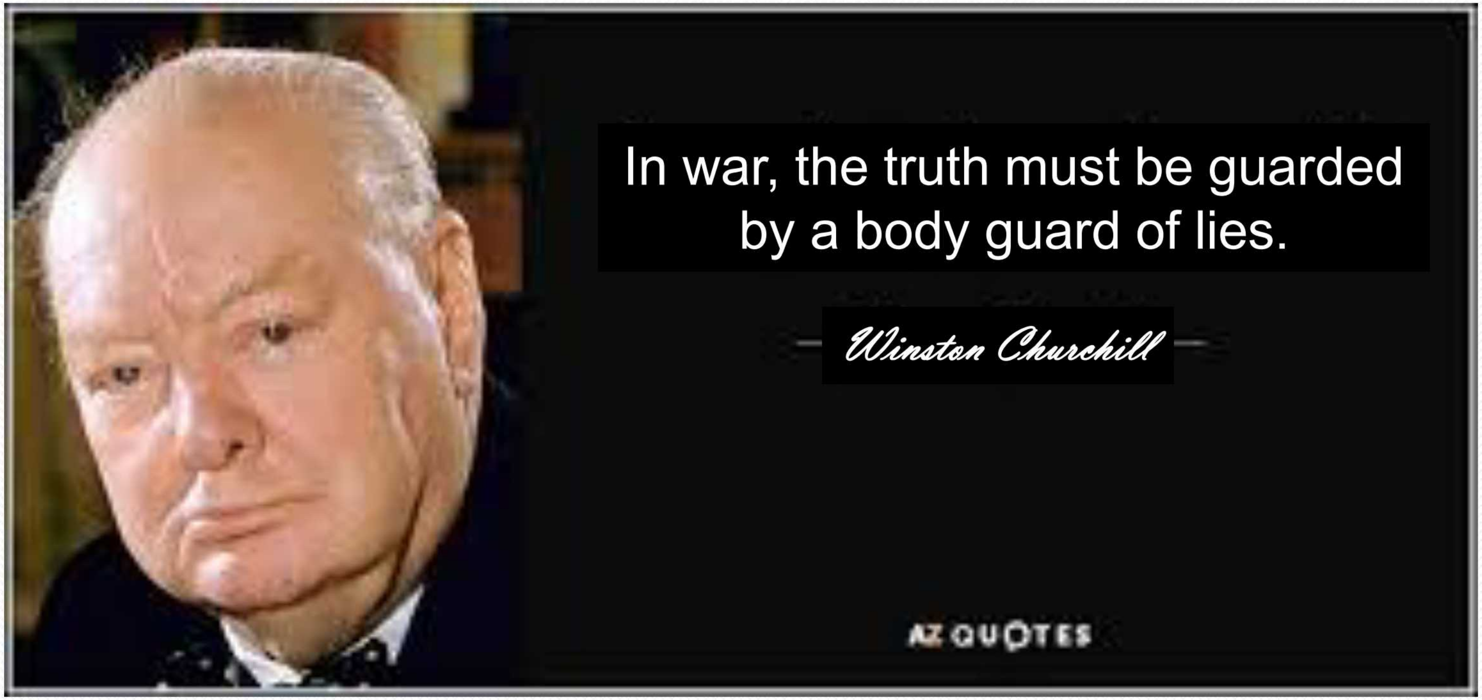
Reinforcement Learning

Supervised Learning



Transfer Learning

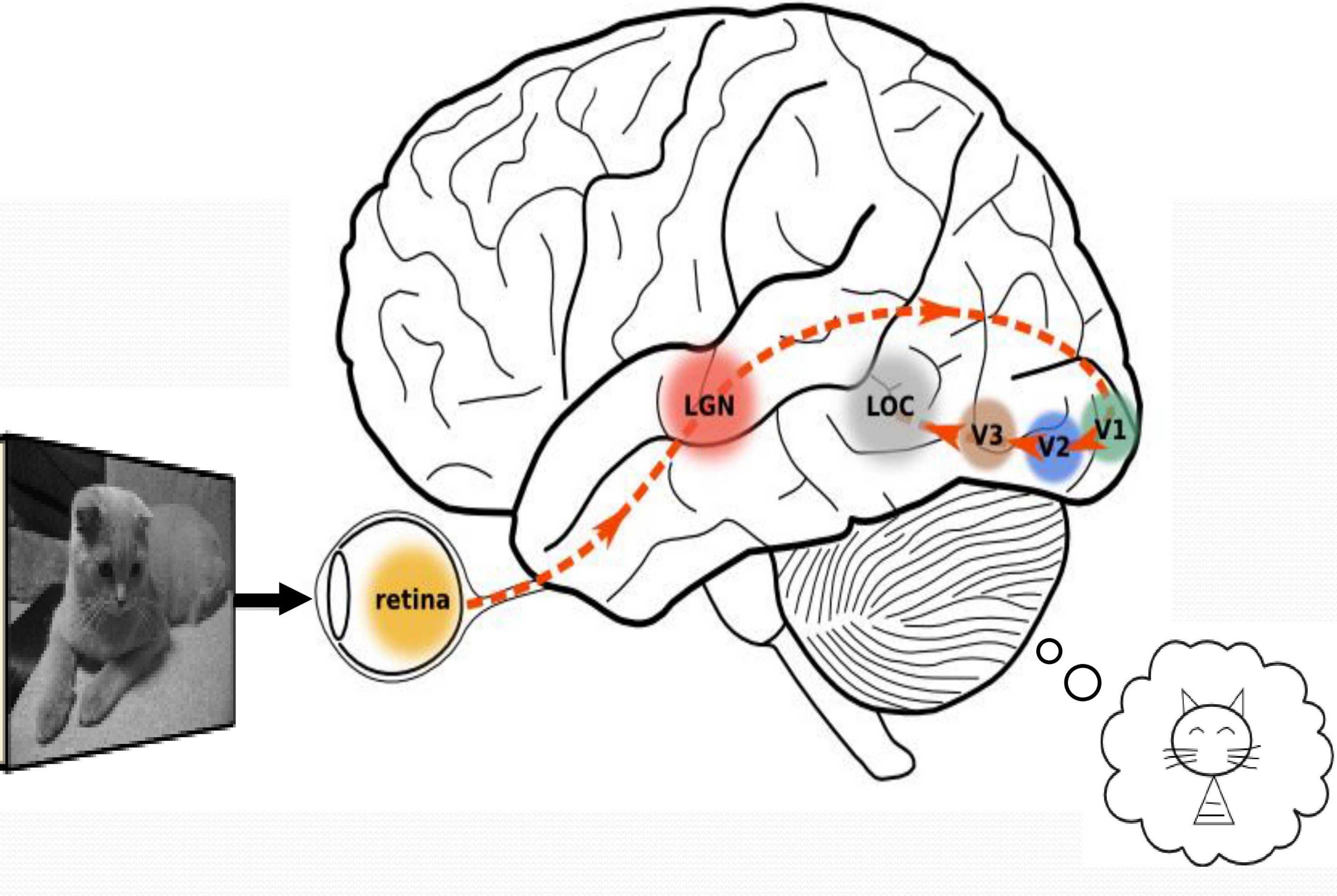
Adversarial Learning



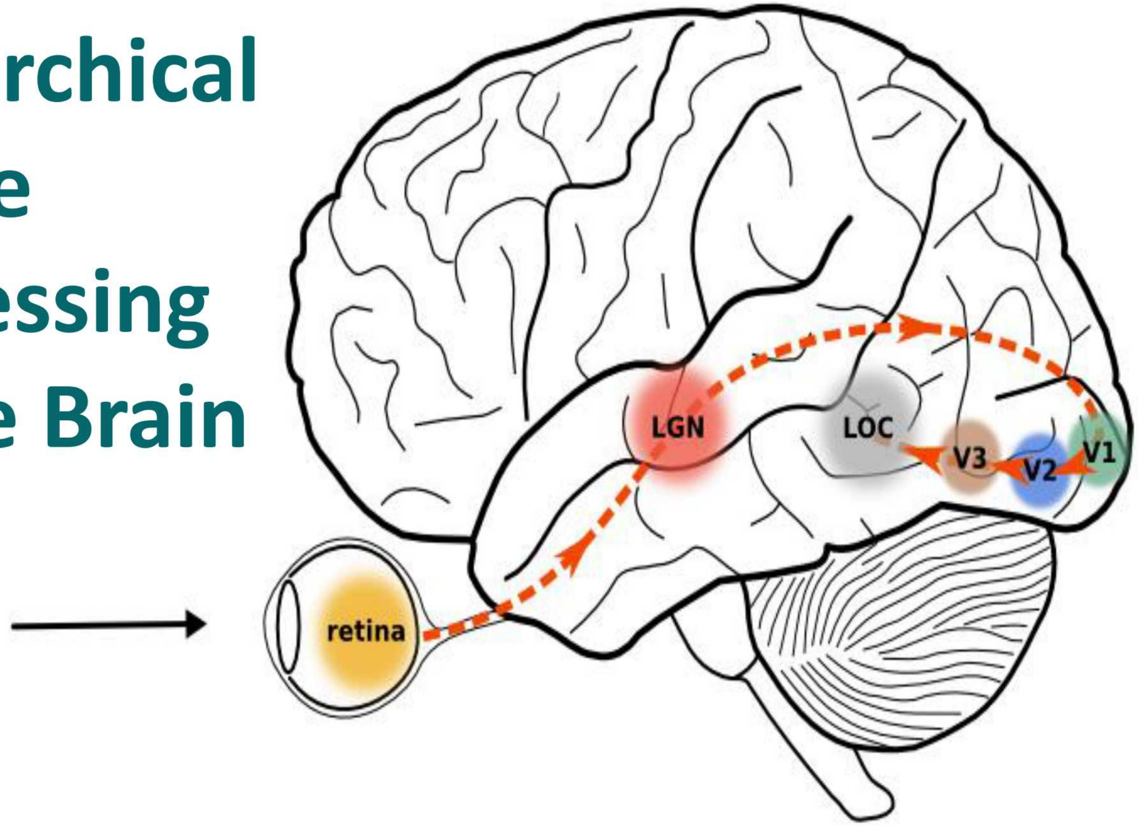
Deep Learning

- Neural Inspiration
- Illustration through a Convolutional Neural Network
- Types of Layers
- Applications
- Stages of development
- Resources

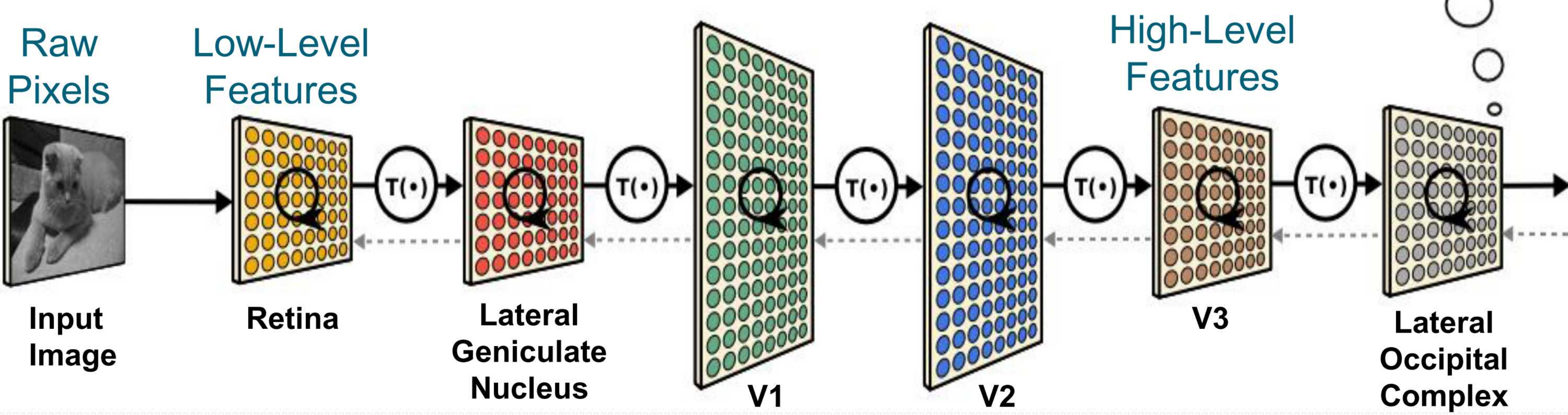
Deep Learning’s Neural Inspiration



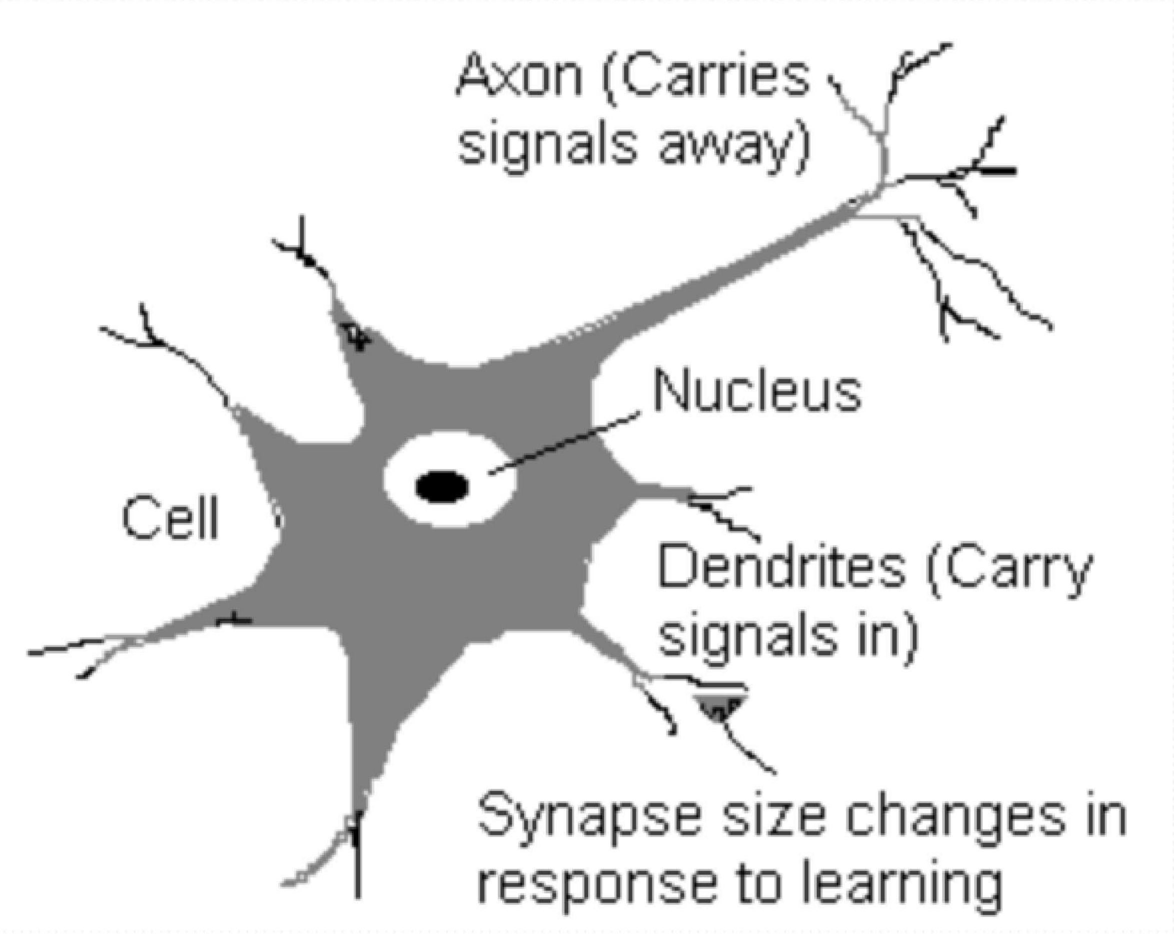
Hierarchical Image Processing in the Brain



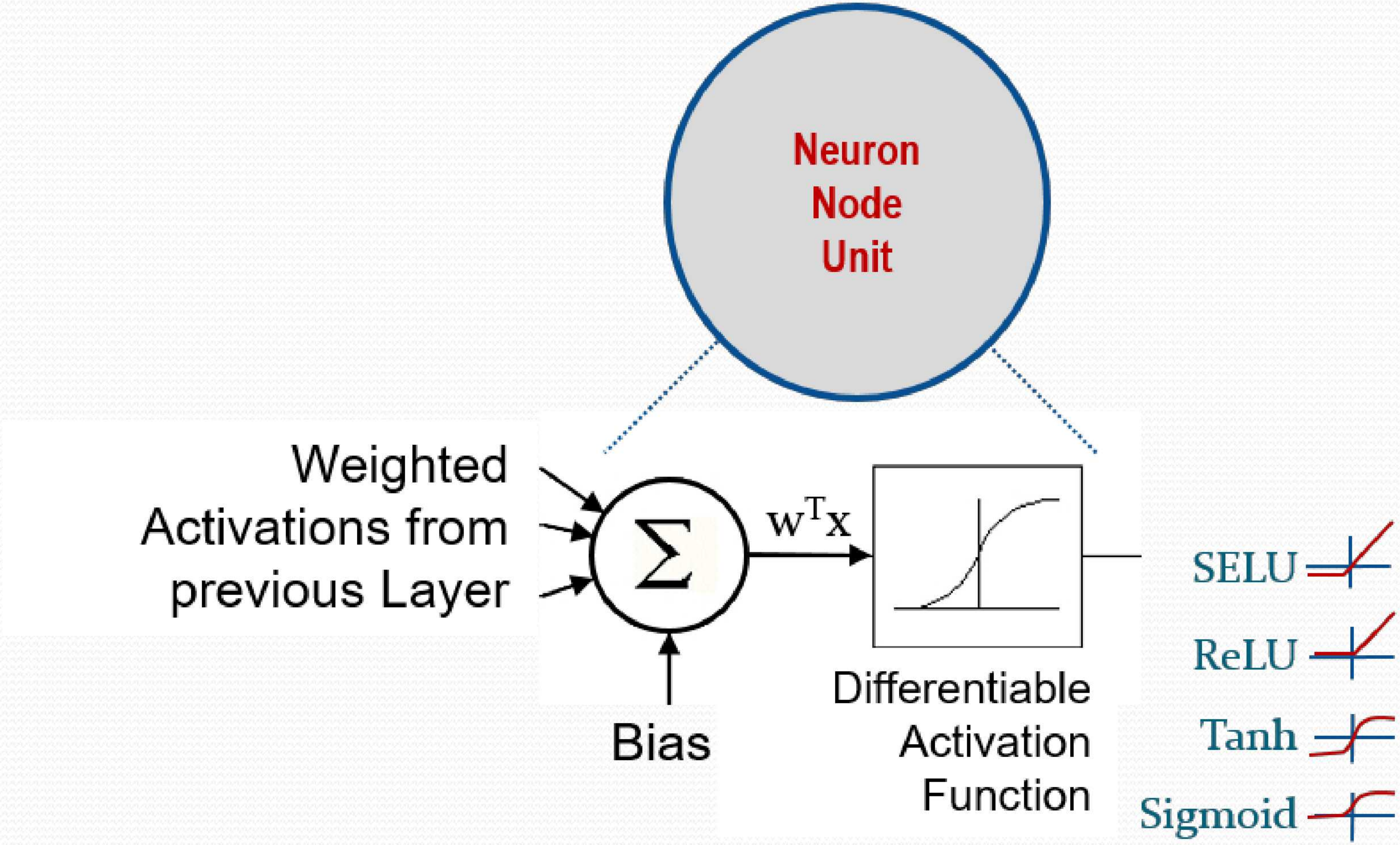
Deep Learning: Neural-Inspired AI



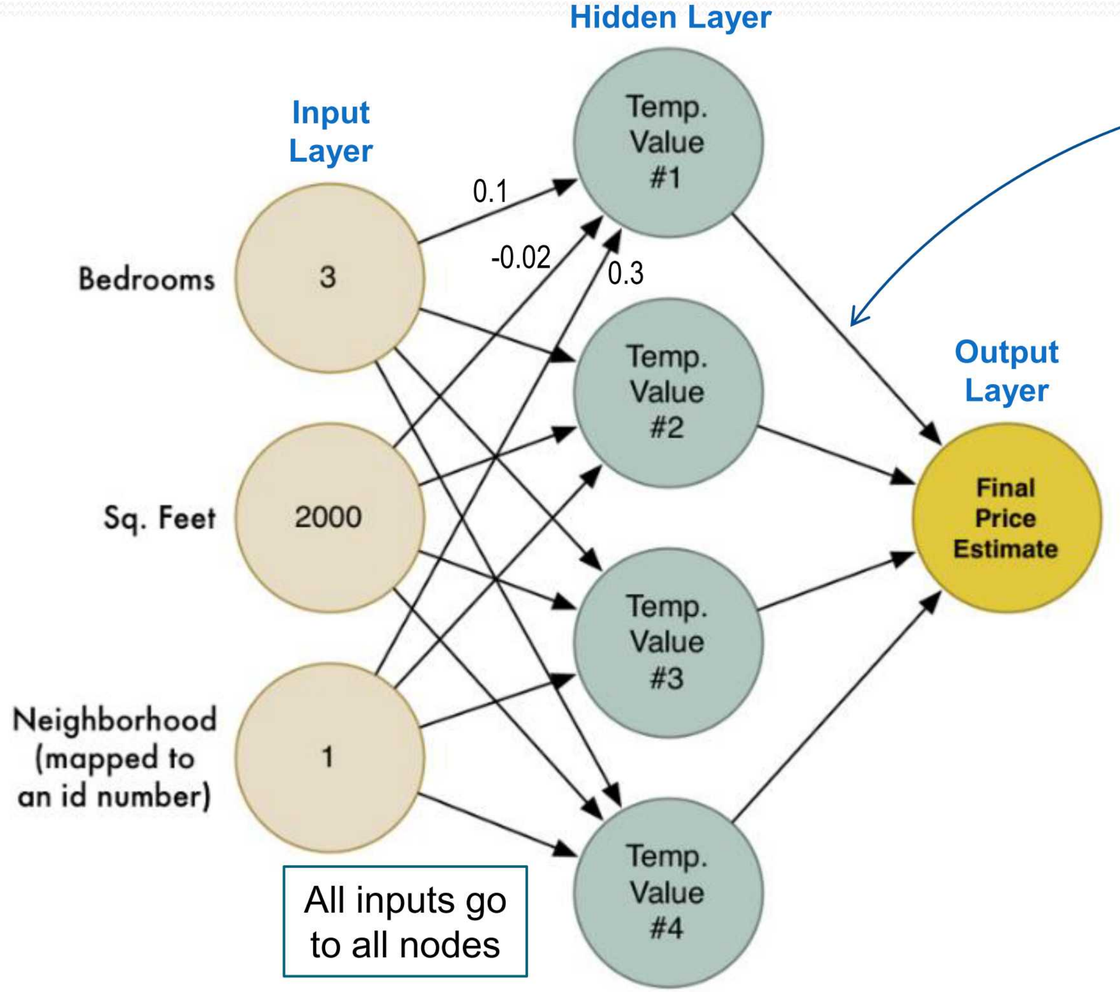
Biological Neuron



“Neuron” for Artificial Neural Networks

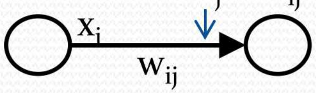


Fully-Connected Feed-Forward Artificial Neural Network



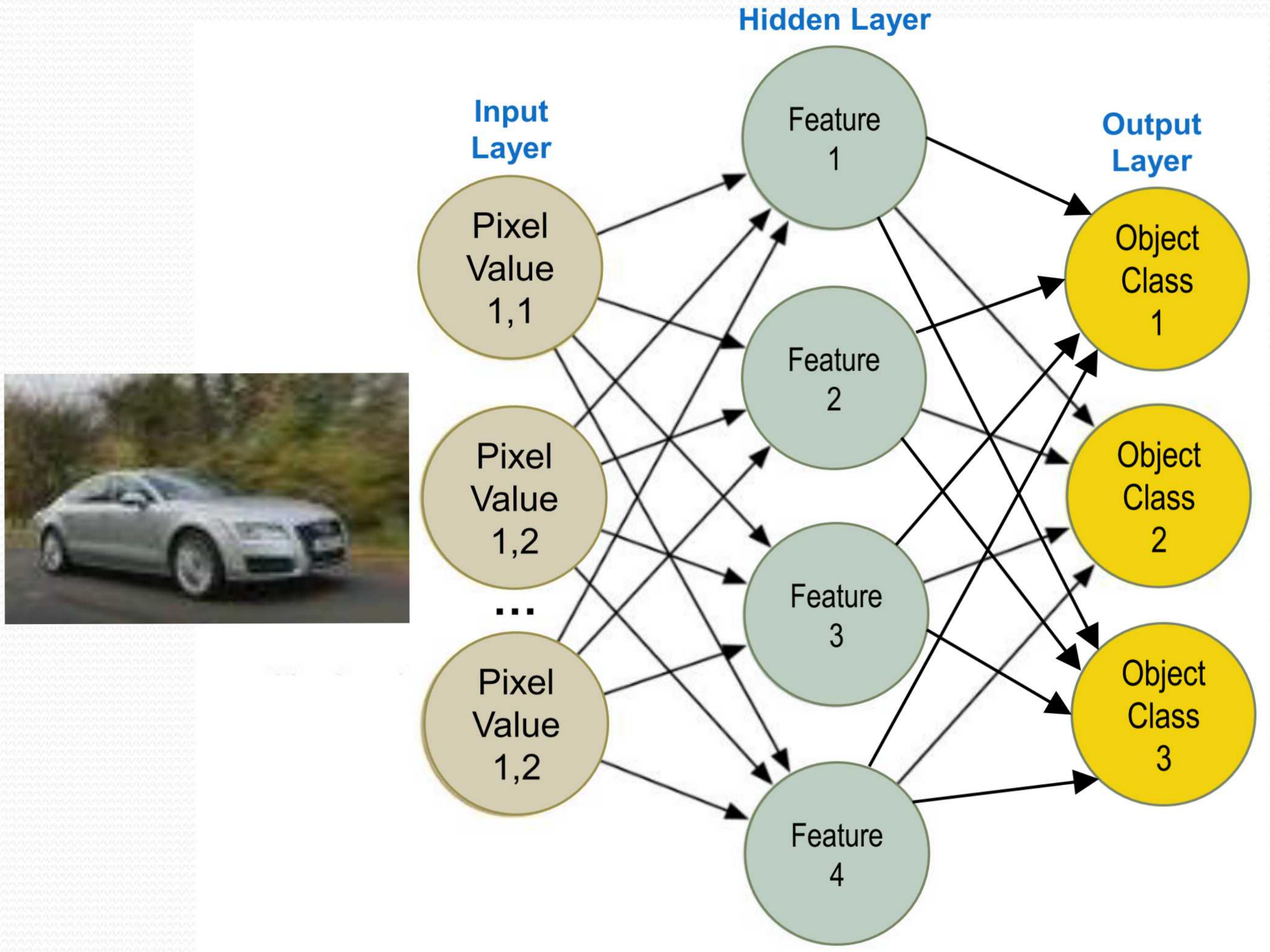
Lines indicate weighted connections between neurons

$z_j = w_{ij} * x_i$

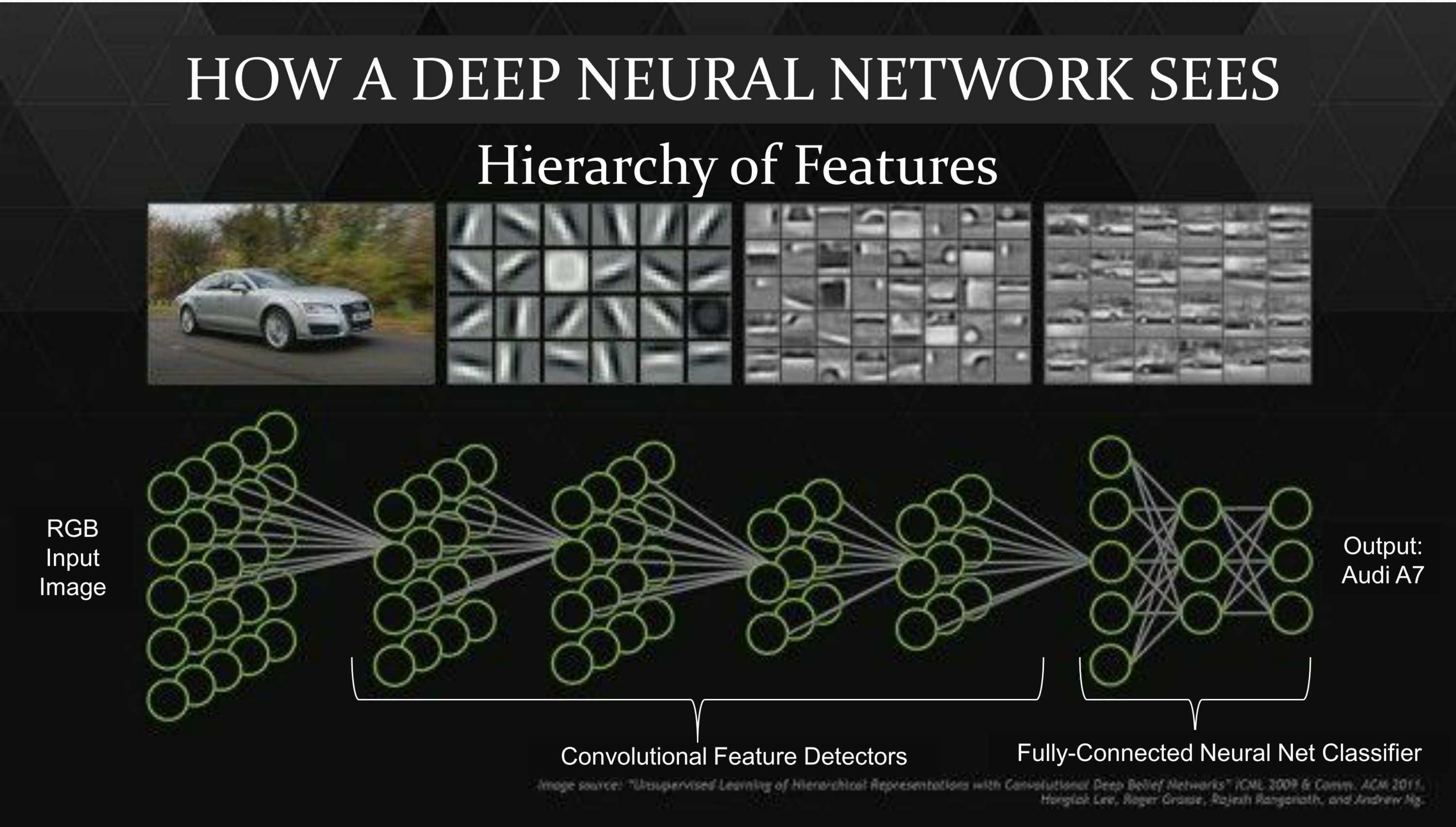


x = activation
w = weight
z = weighted activation

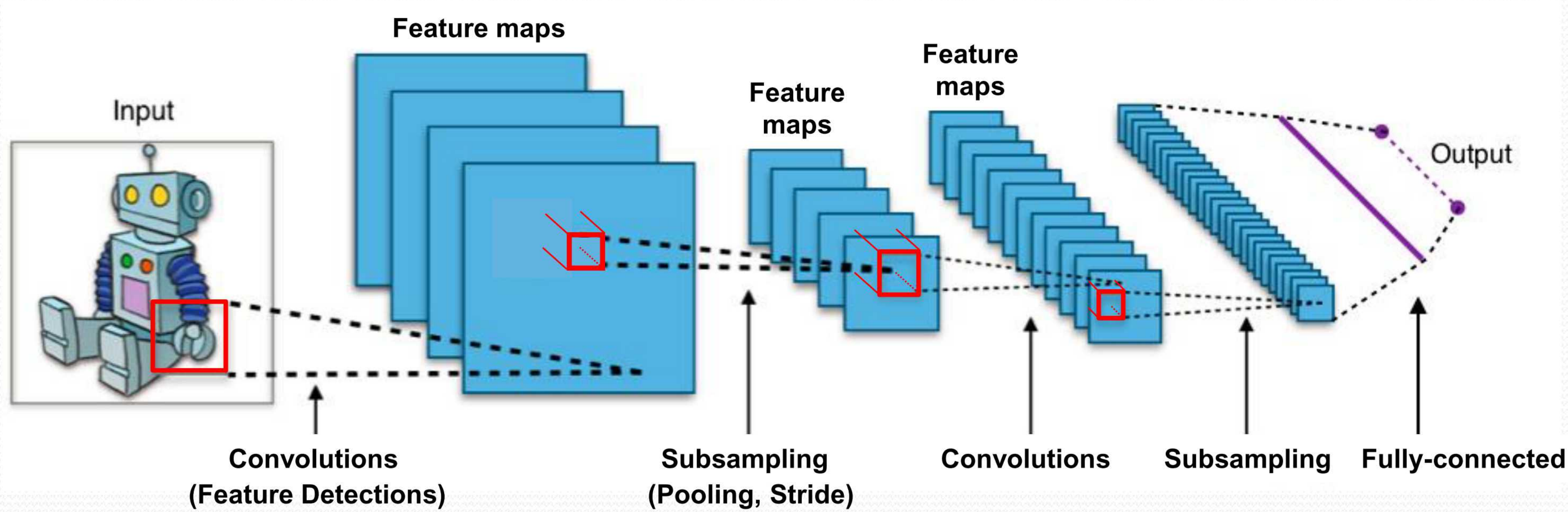
Fully-Connected Feed-Forward Artificial Neural Network



Convolutional Neural Network (CNN)



CNN Operation



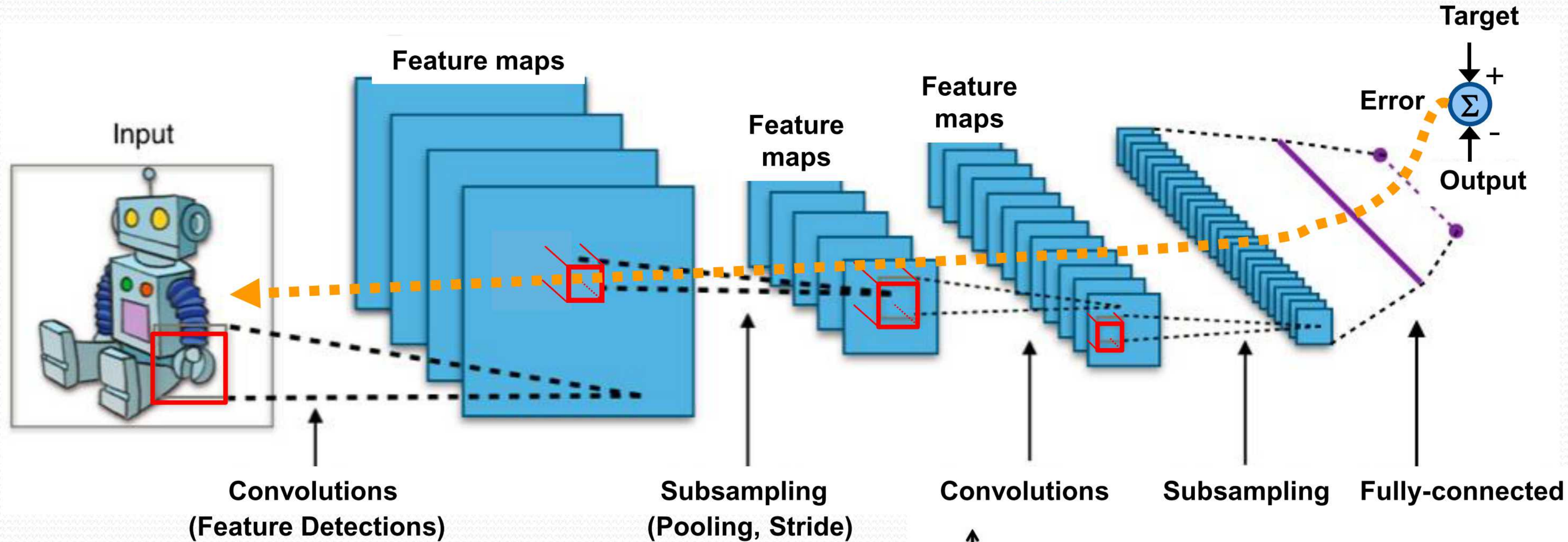
- Capture (Training) or Detect (Testing) spatial structure (features)
  - Convolution is used to find features in signals (template matching)

Let f be the signal and g be a feature template/filter/kernel

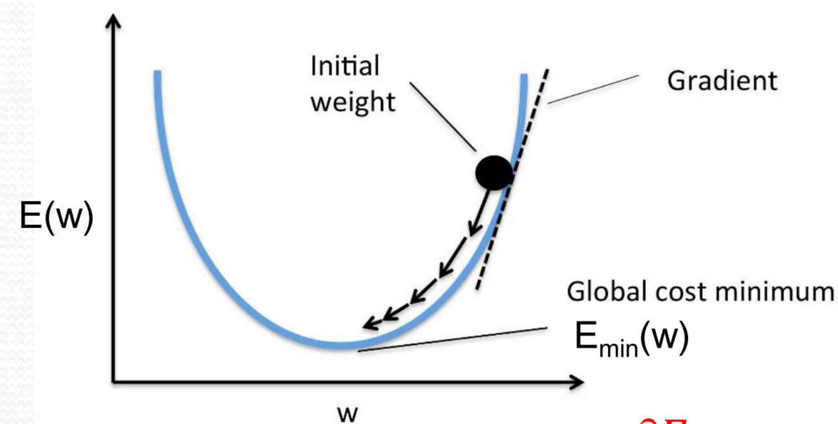
$$(f^*g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau$$

$$(f^*g)(t) = \int_{m=-M}^{M} f(m)g(t-m)$$

CNN Training



- Convolution filters are the weights in a CNN that can be trained.
- Filter values (weights) are initialized with random values and update via back-propagation.

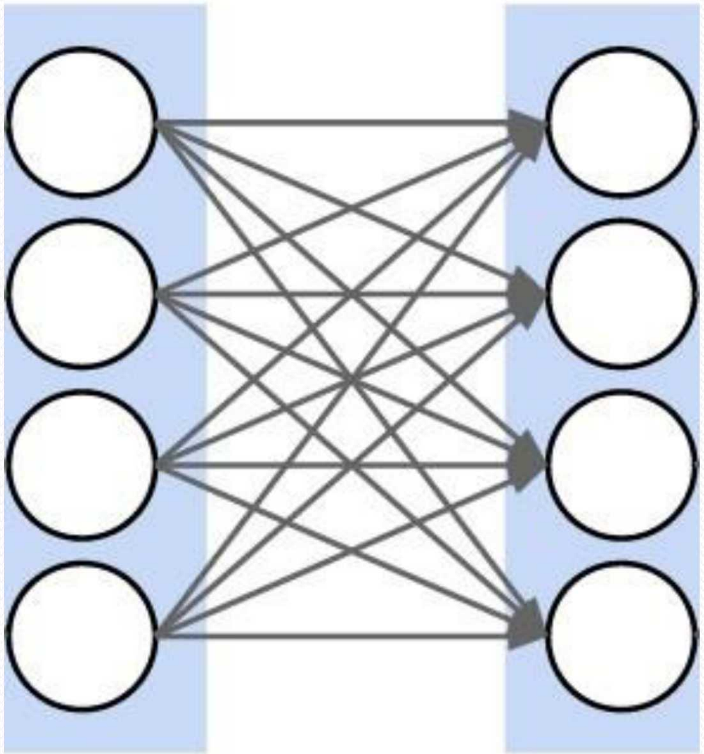


$$w(t+1) = w(t) - \eta\Delta\frac{\partial E}{\partial w(t)}$$

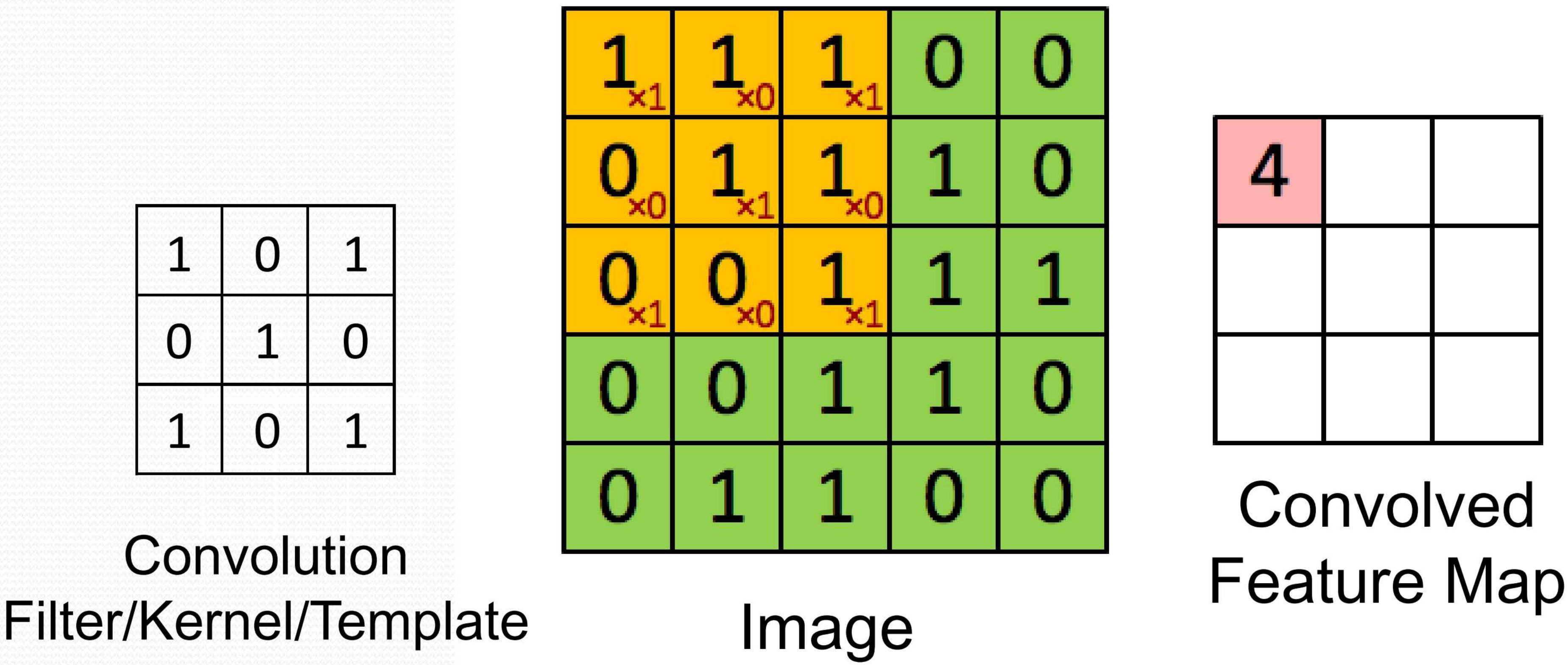
Deep Neural Network (DNN) Layers

- Fully Connected
- Convolutional
- Pooling
- Batch Normalization
- Recurrent
- Activation
- Output
  - Softmax
  - Logistic Sigmoid

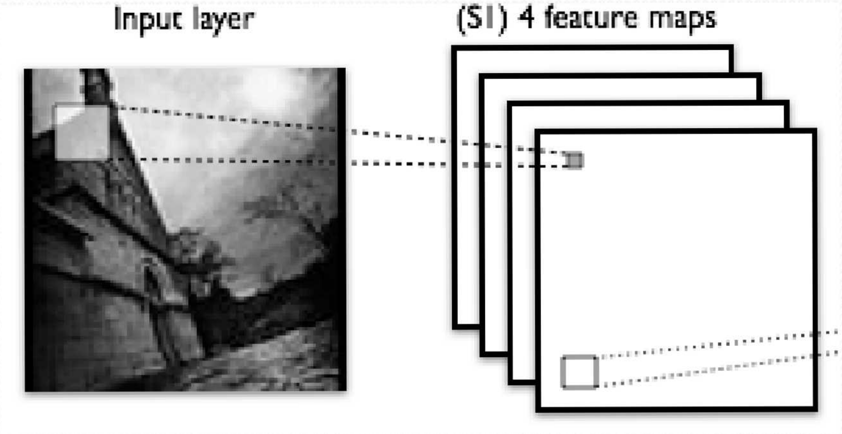
Fully Connected Layer

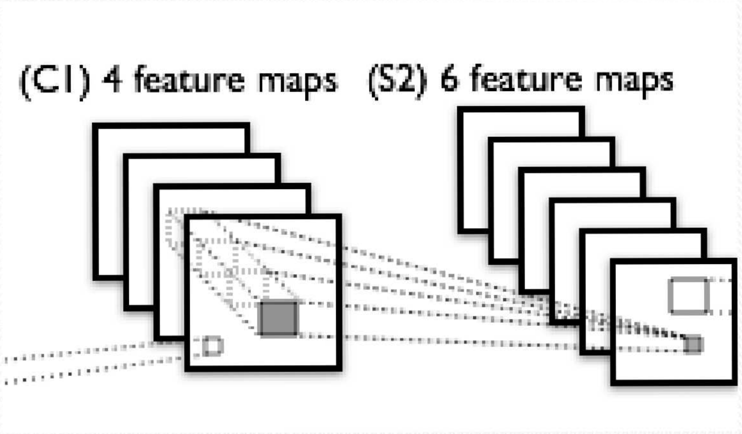


Convolutional Layer

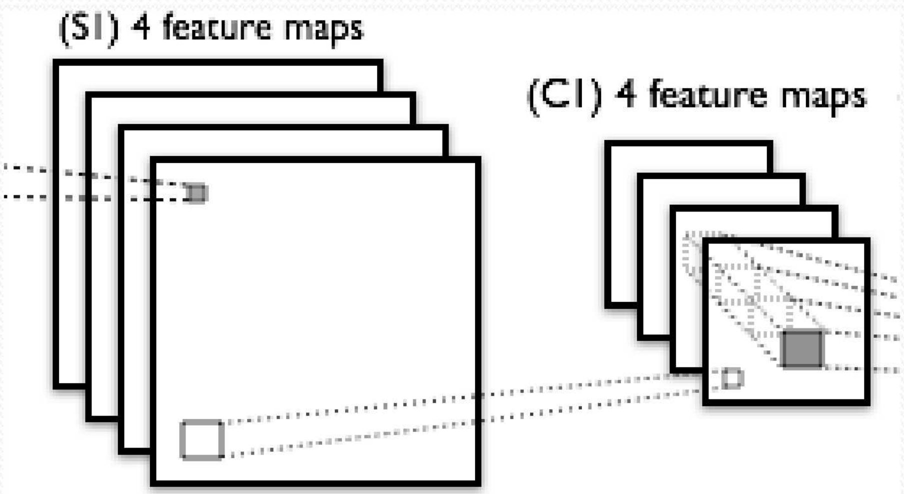


Convolutional Layer

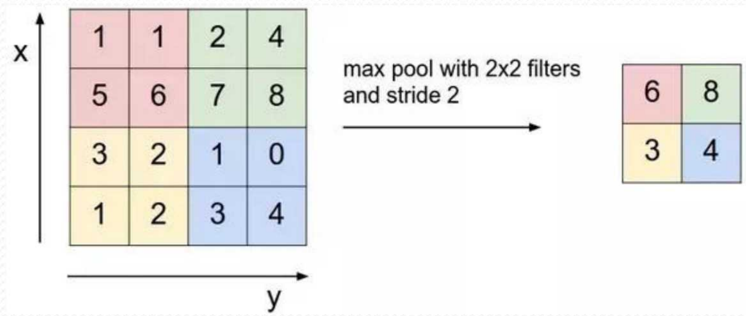




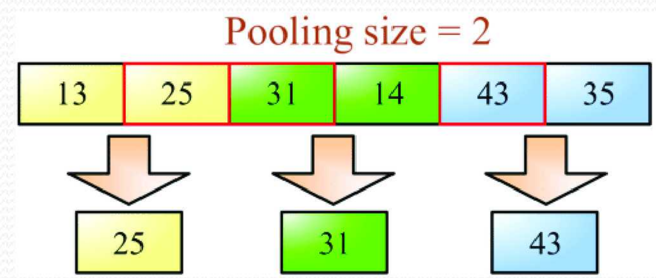
Pooling Layer



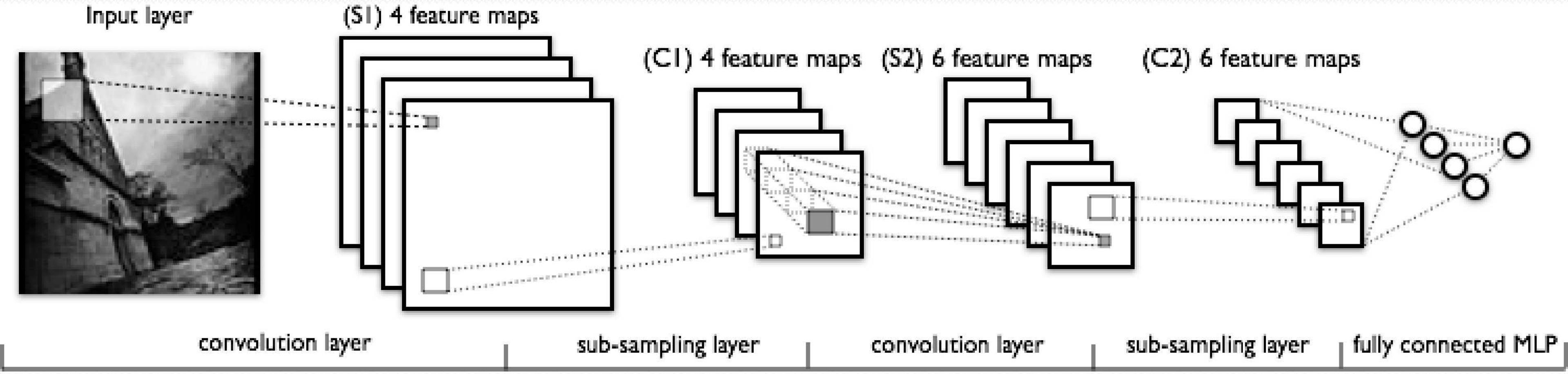
2-D:



1-D:



Full CNN



Input layer
(S1) 4 feature maps
(C1) 4 feature maps
(S2) 6 feature maps
(C2) 6 feature maps
convolution layer
sub-sampling layer
convolution layer
sub-sampling layer
fully connected MLP

Batch Normalization Layer

- Reduces **Internal Covariate Shift**
  - Change in the distribution of network activations due to the change in network parameters during training.
- Network training converges faster with whitened inputs
  - Zero means, unit variances, and decorrelated.

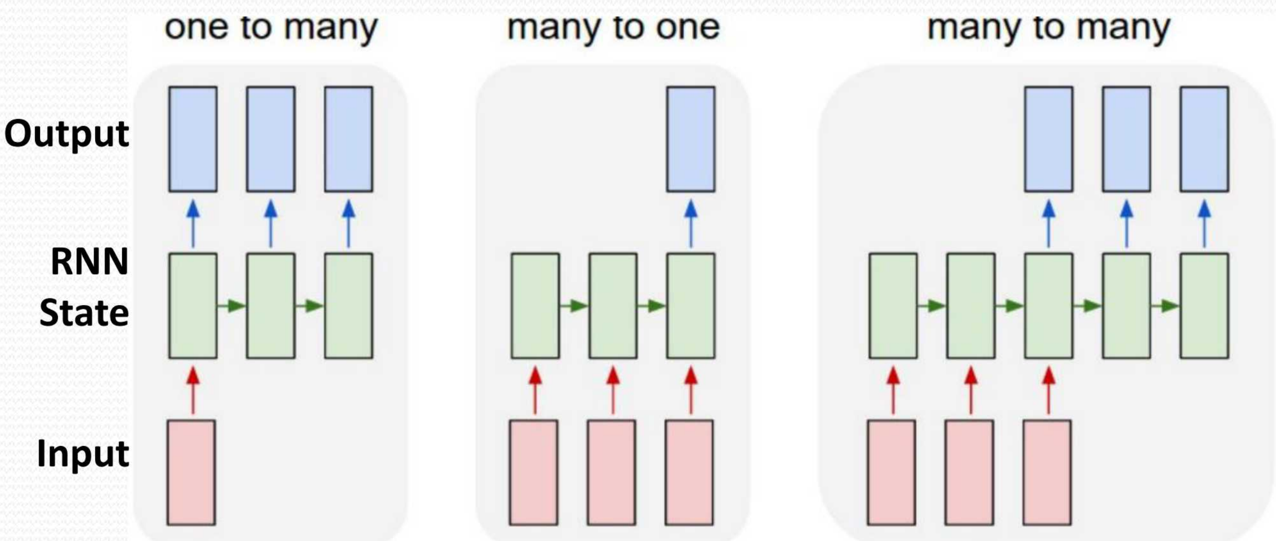
Whitening $$\hat{x}^k = \frac{x^k - E[x^k]}{\sqrt{Var[x^k]}}$$ **Normalization** $$y^k = \gamma^k \hat{x} + \beta^k$$

- Reduces the dependence of gradients on the scale of the parameters or their initial values.
- Regularizes the model and reduces the need for dropout and other regularization techniques.
- Allows use of saturating nonlinearities (e.g., sigmoid, tanh) and higher learning rates.

Recurrent Layer

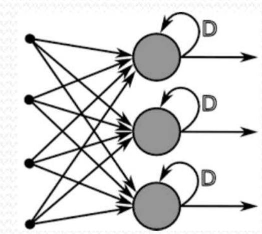
Long Short-Term Memory (LSTM)
Gate Recurrent Unit (GRU)

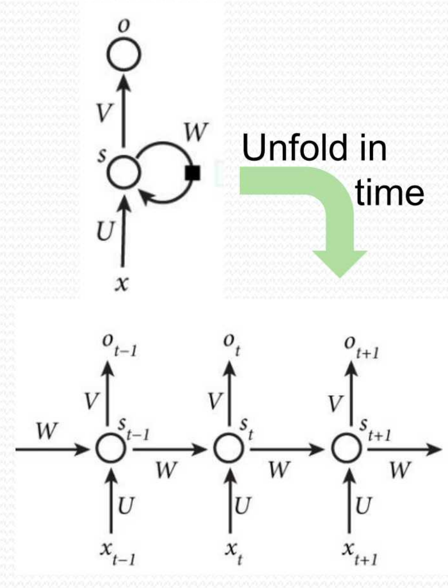
- Learning patterns in sequences



Each rectangle is a vector and arrows represent functions (e.g. matrix multiply).
No constraint on lengths of sequences because the recurrent transformation (green) is fixed and can be applied as many times as desired.

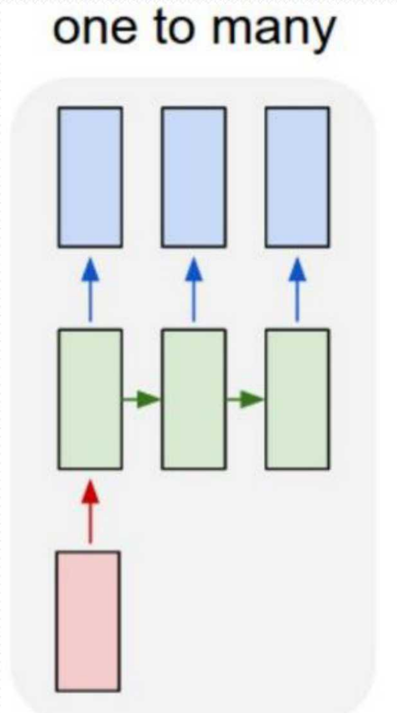
3 RNN Nodes



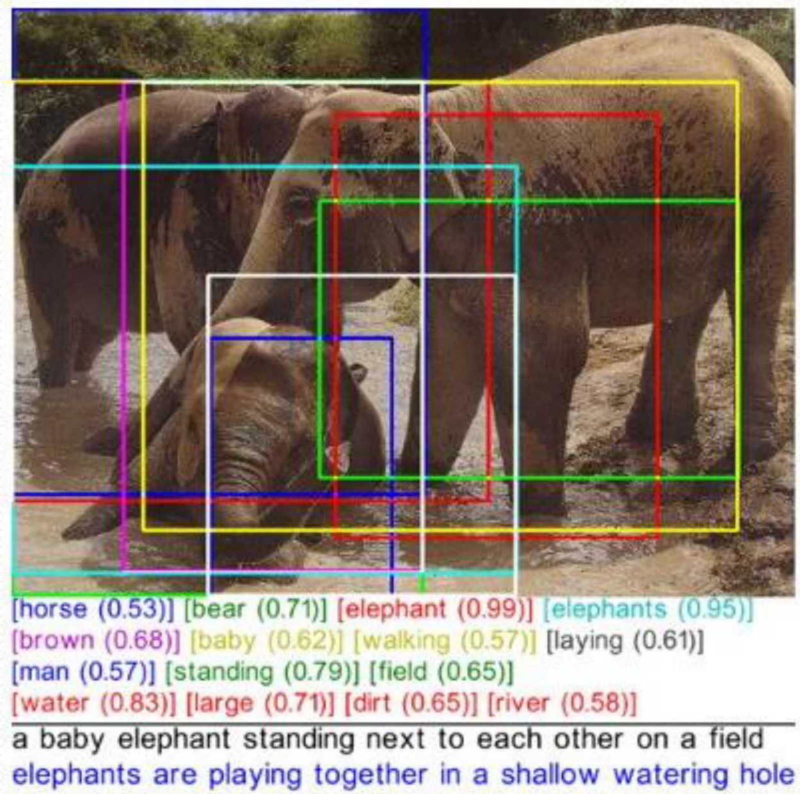


A. Karpathy, The Unreasonable Effectiveness of Recurrent Neural Networks, May 21, 2015
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

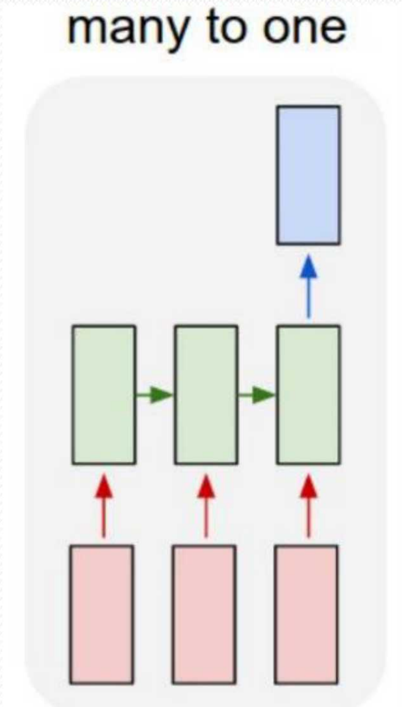
One to Many RNN



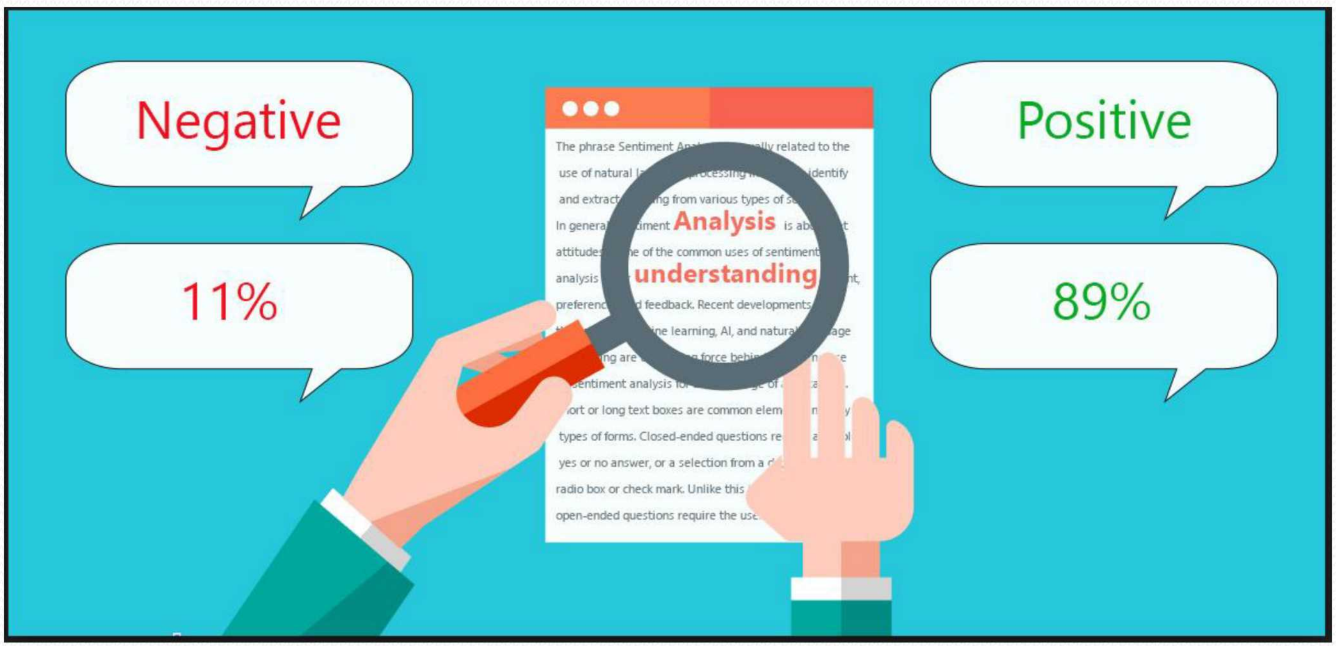
Sequence output (e.g. image captioning takes an image and outputs a sentence of words)



Many to One RNN

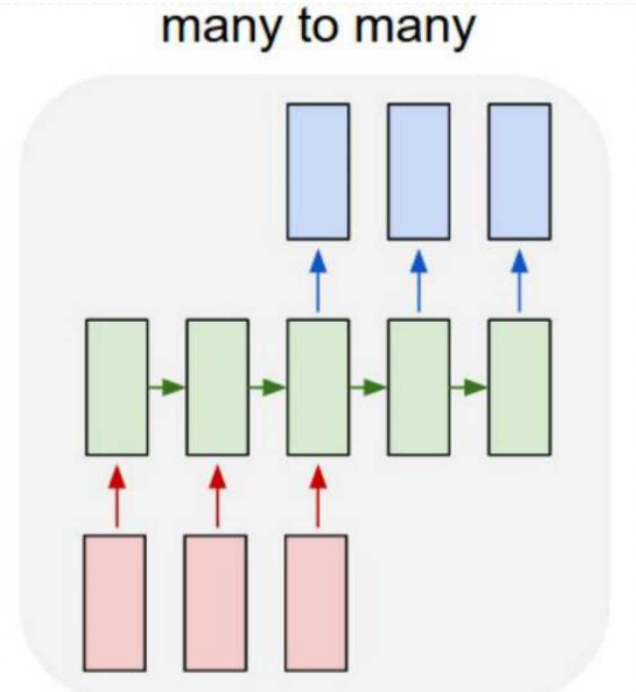


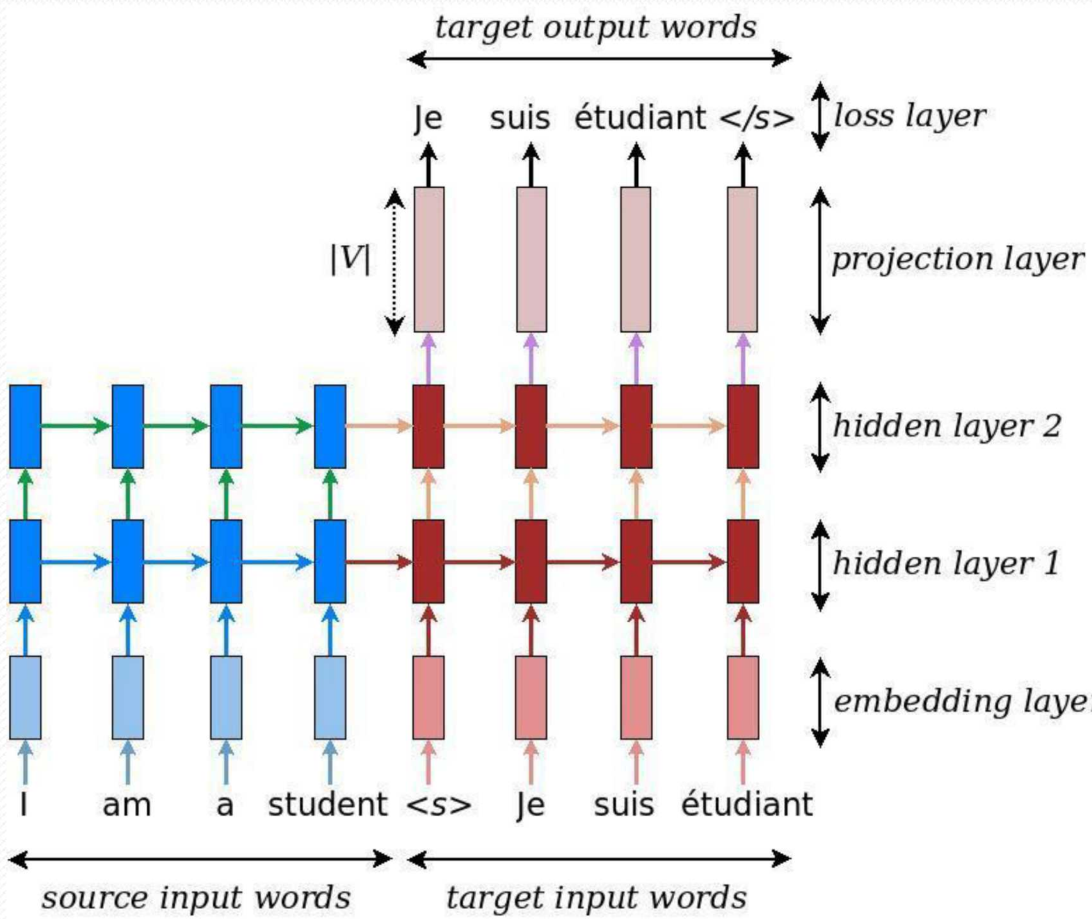
Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment).



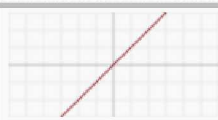
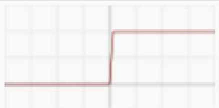
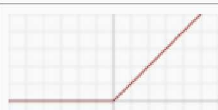
Many to Many RNN

Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French).





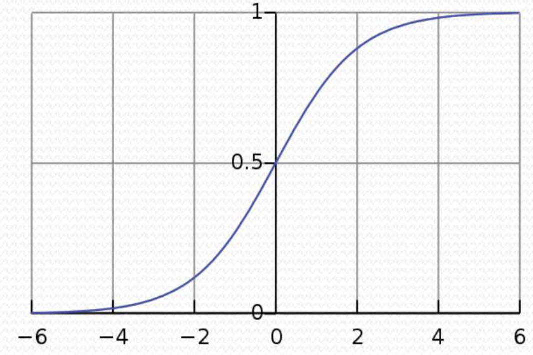
Activation Layer

Name	Plot	Equation	Derivative
Linear		$f(x) = x$	$f'(x) = 1$
Step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic Sigmoid		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
tanh		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
arctan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear, ReLU		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parametric ReLU		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Softplus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Output Layer

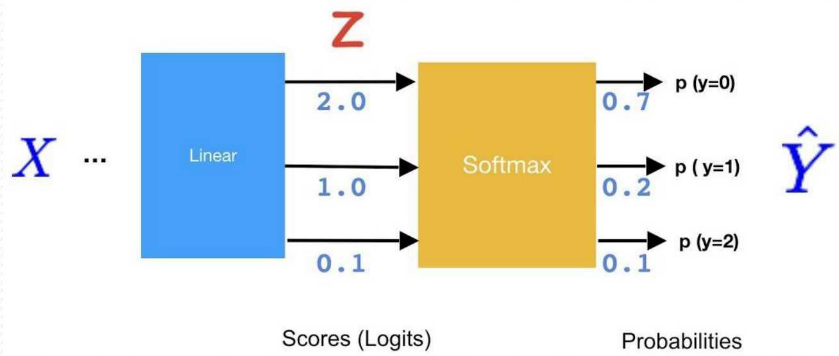
- **Logistic (Squashing) Function**
  - Output is between 0 and 1

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

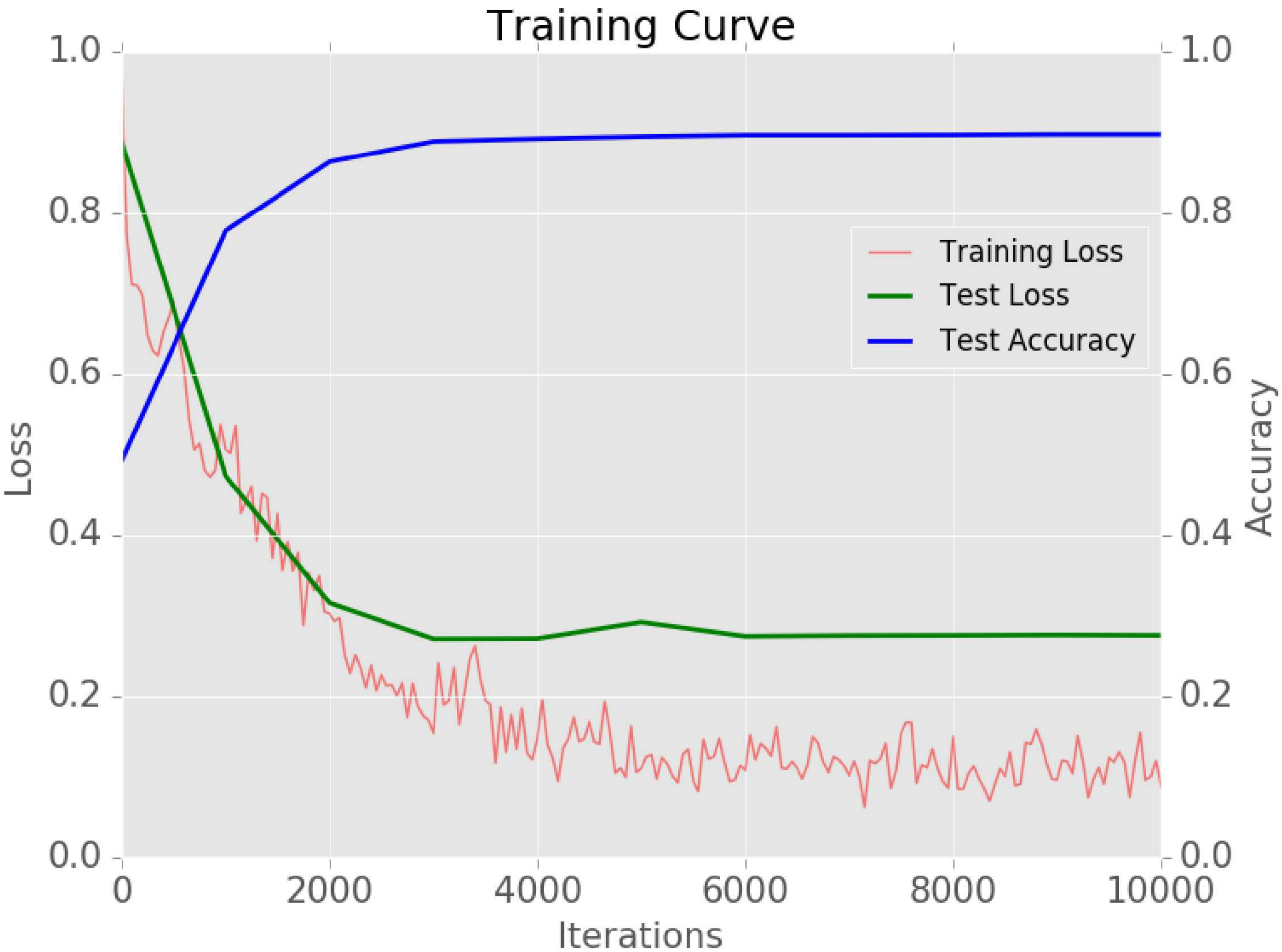


- **Softmax (Normalized Exponential) Function**
  - Probability distribution over K different possible outcomes
    - Each output is between 0 and 1
    - All outputs add up to 1

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \dots, K$$



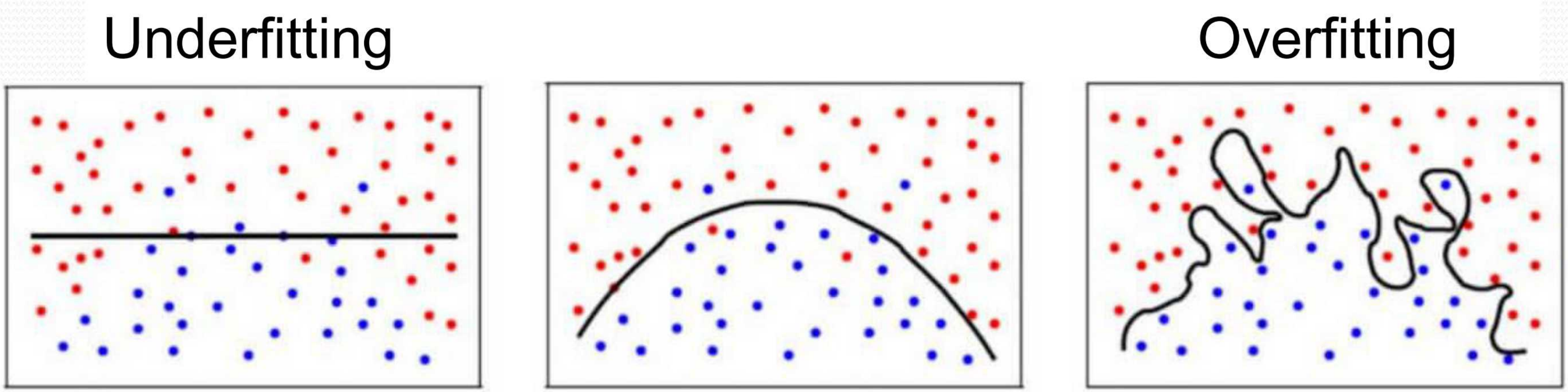
Supervised Training



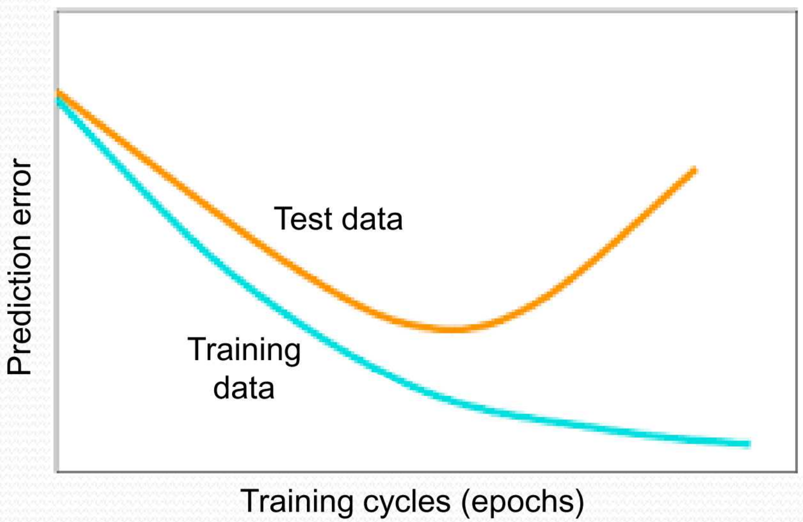
Training Terms

- **Epoch**: Pass through all training samples
- **Minibatch**: Subset of training samples
- **Iteration**: Pass through 1 minibatch
- **Learning Rate**: Size of training step
- **Loss**: Classification error
- **Accuracy**: Percent of correct classifications

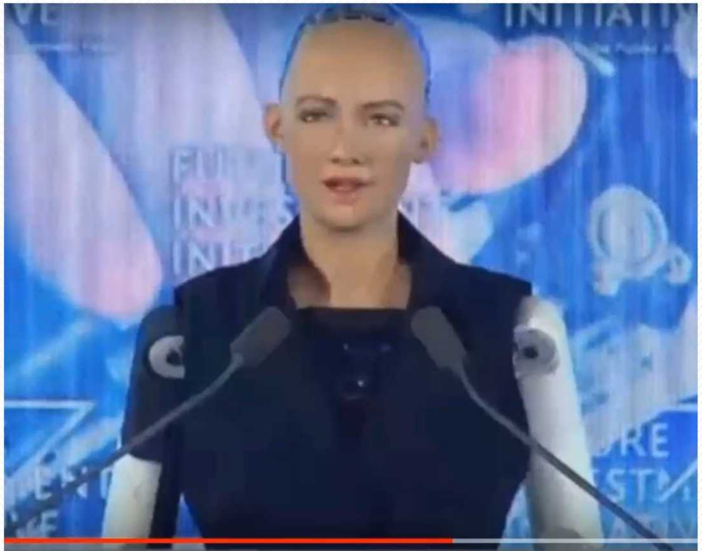
Generalization, Overfitting



- Overfitting
  - Over-capacity
  - Over-training
  - Limited data
  - Unbalanced data distributions
- Generalization
  - Data augmentation
  - Regularization
  - Early stopping

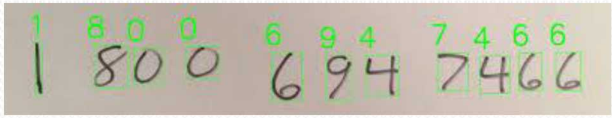


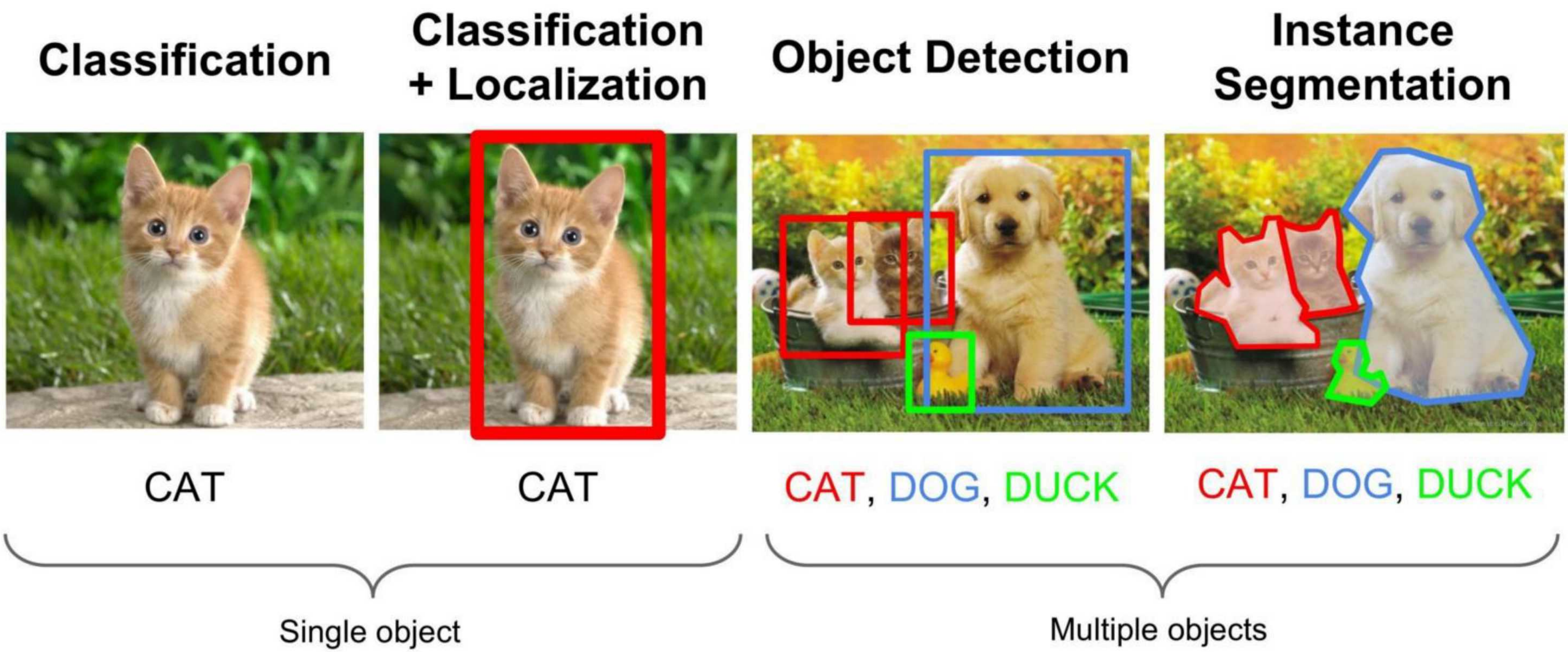
Applications/Types of DNNs



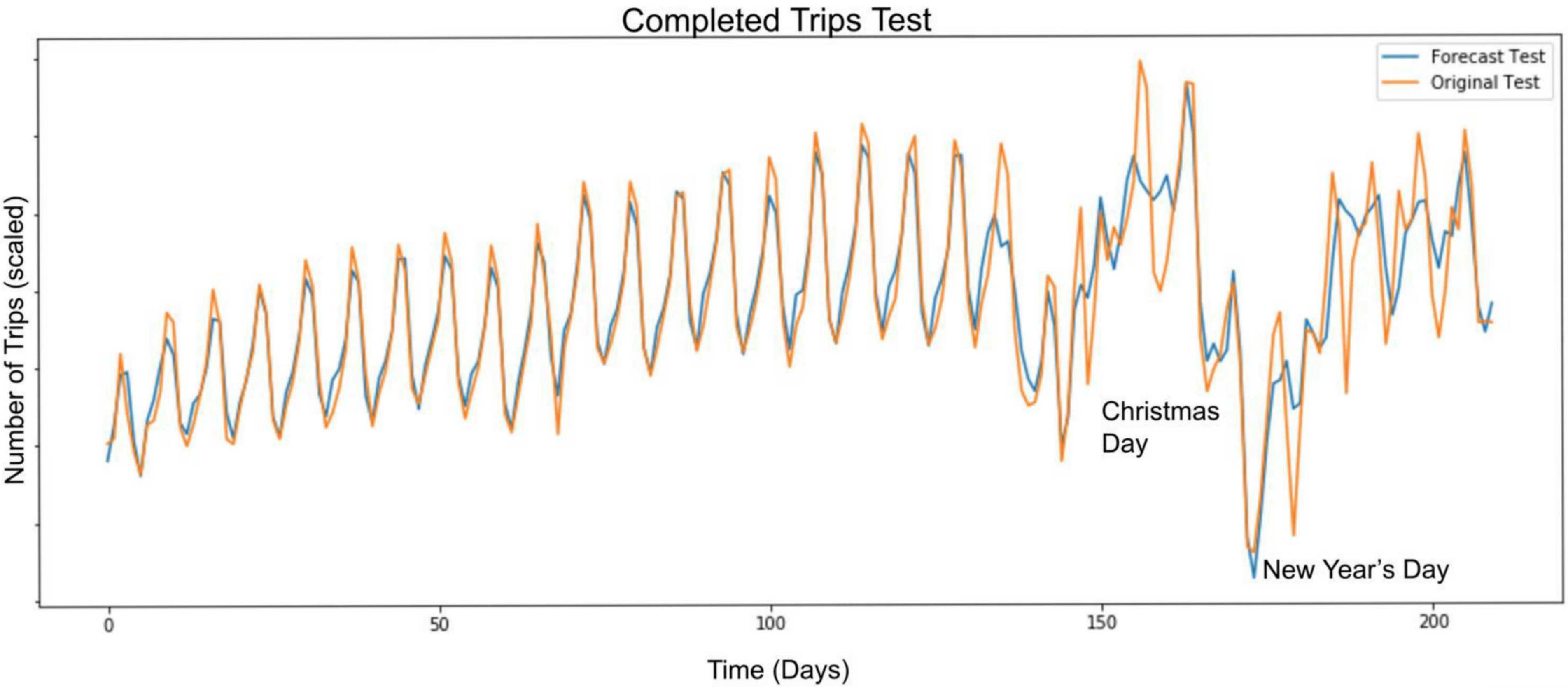
Classification, Segmentation

- Object, Speech, Character Recognition





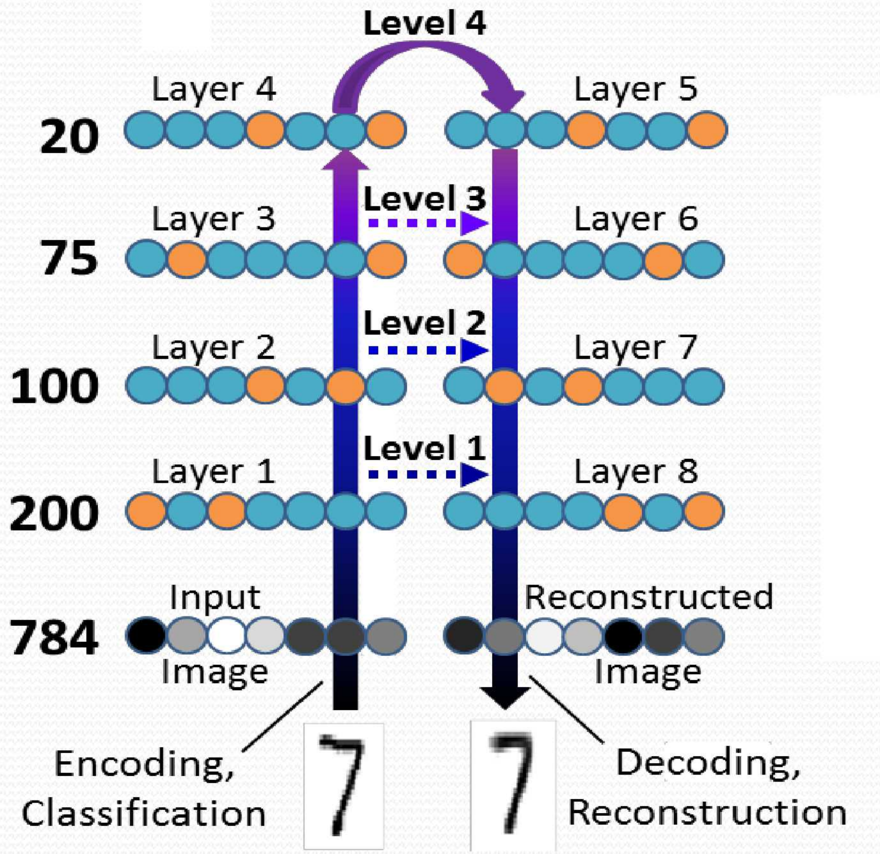
Prediction



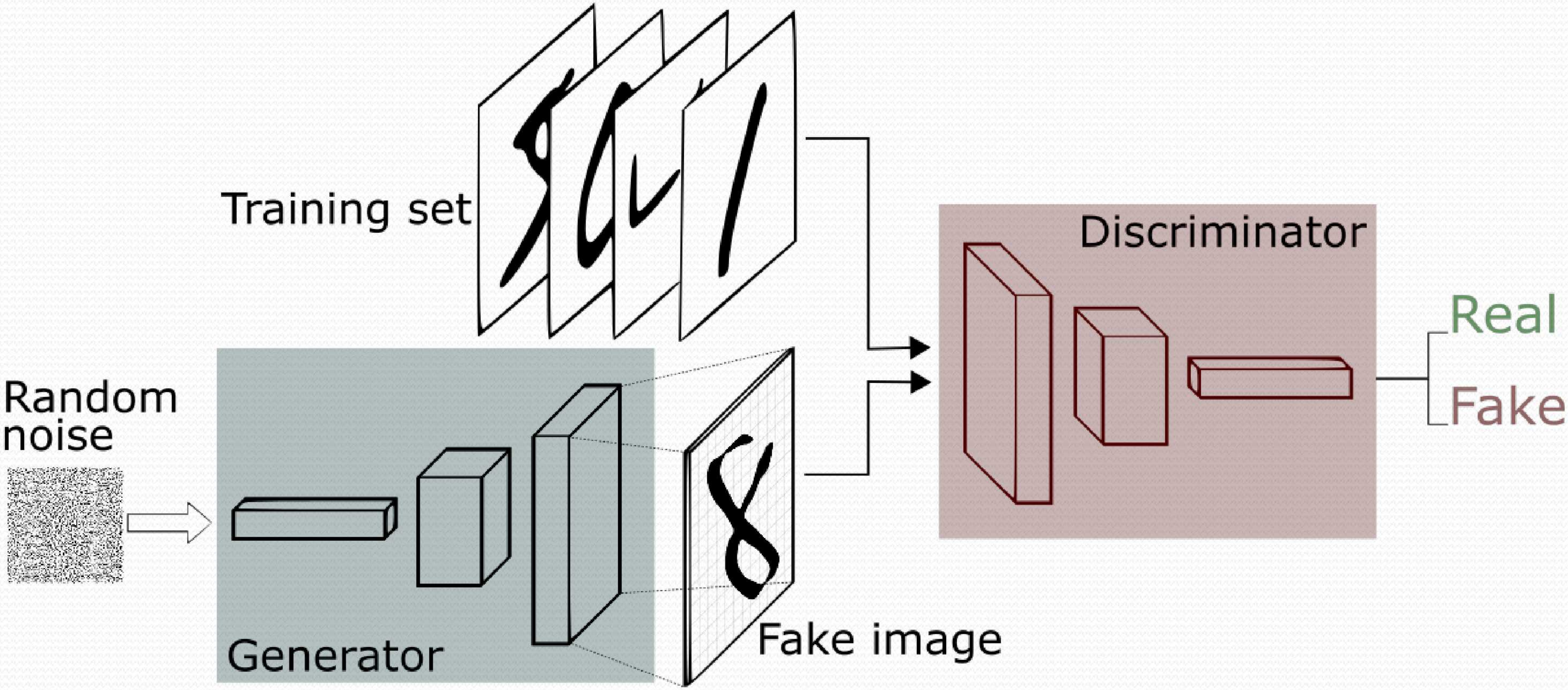
Completed Trips Test
Forecast Test
Original Test
Number of Trips (scaled)
Time (Days)
0
50
100
150
200
Christmas Day
New Year's Day

Generation

Autoencoder (AE) Neural Network

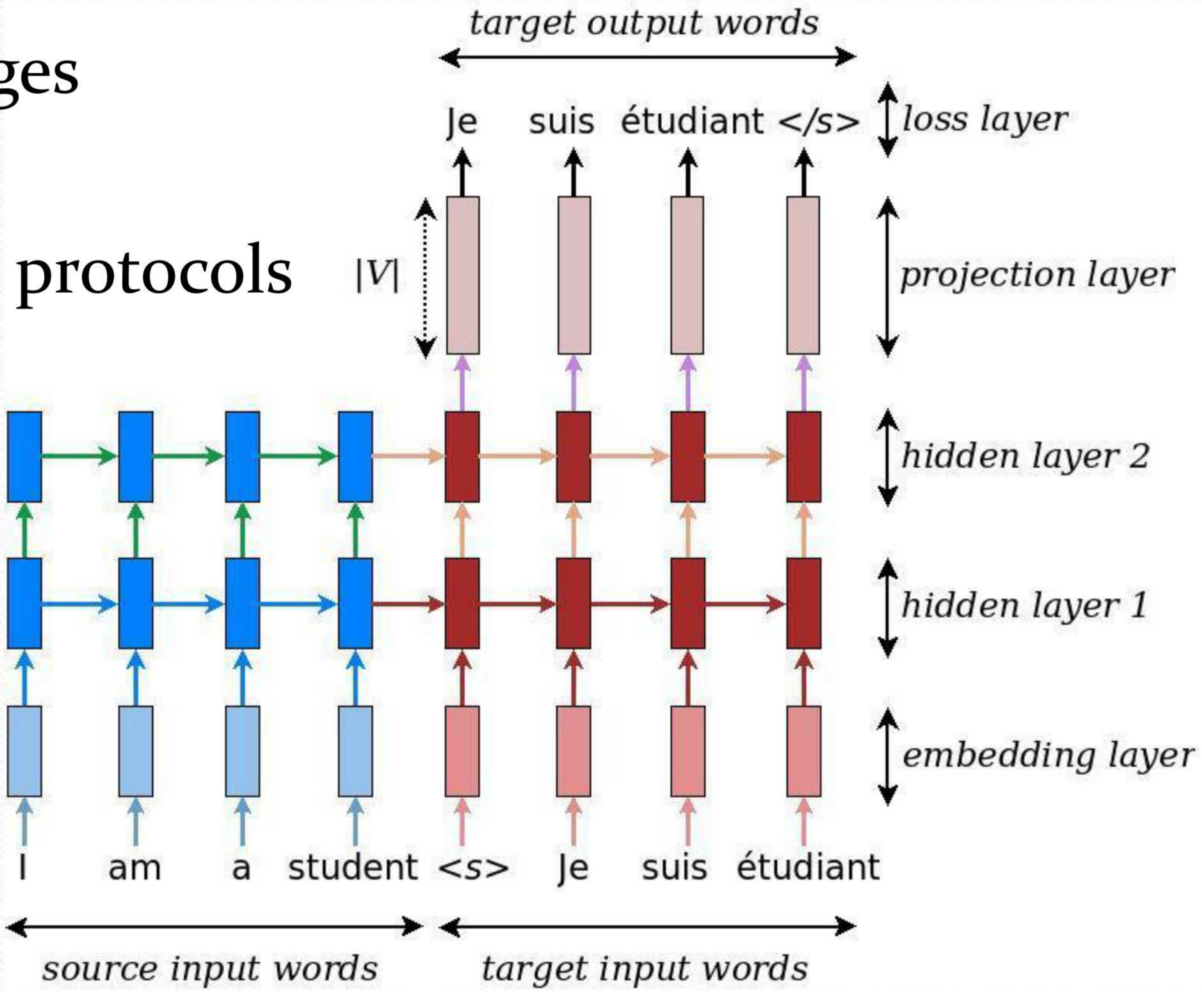


Generative Adversarial Network (GAN)



Translation

- Written Languages
- Source Code
- Communication protocols



Stages of Deep Learning

- Collect Data
  - Understand
  - Filter
  - Normalize
    - Size (pad, crop)
    - Magnitude
  - Augment
- Design DNN
  - Layers
  - Elements/Layer
  - Output
- Train
- Test/Evaluate
  - Confusion Matrix
  - Misclassified samples
- REPEAT

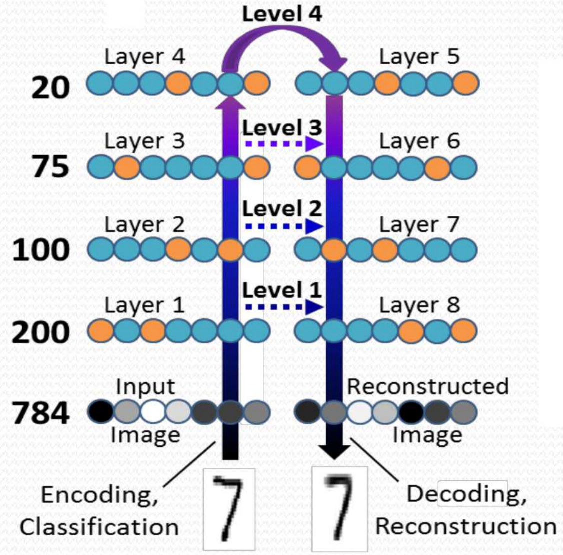
ImageNet Challenge

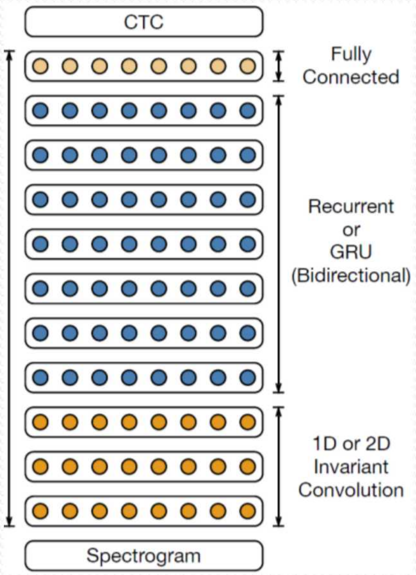
IMAGENET

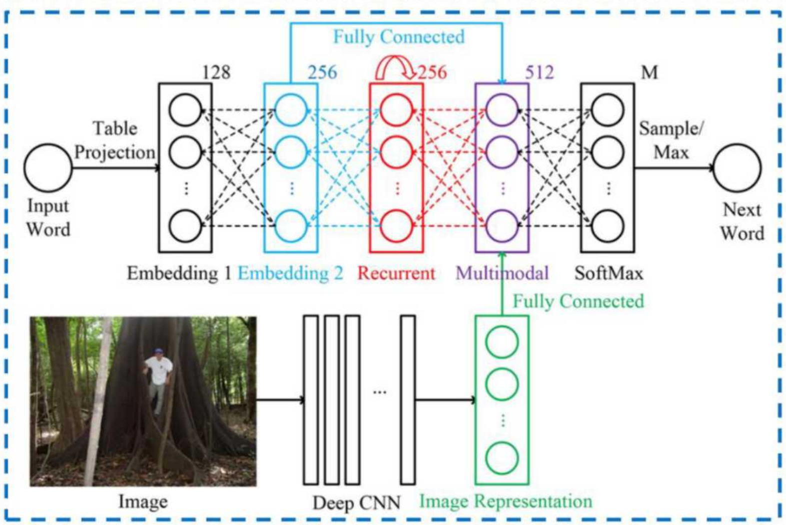
- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

Stages of Deep Learning

- Collect Data
  - Understand
  - Filter
  - Normalize
    - Size
    - Magnitude
  - Augment
- Design DNN
  - Layers
  - Elements/Layer
  - Output
- Train
- Test/Evaluate
  - Confusion Matrix
  - Misclassified samples
- REPEAT

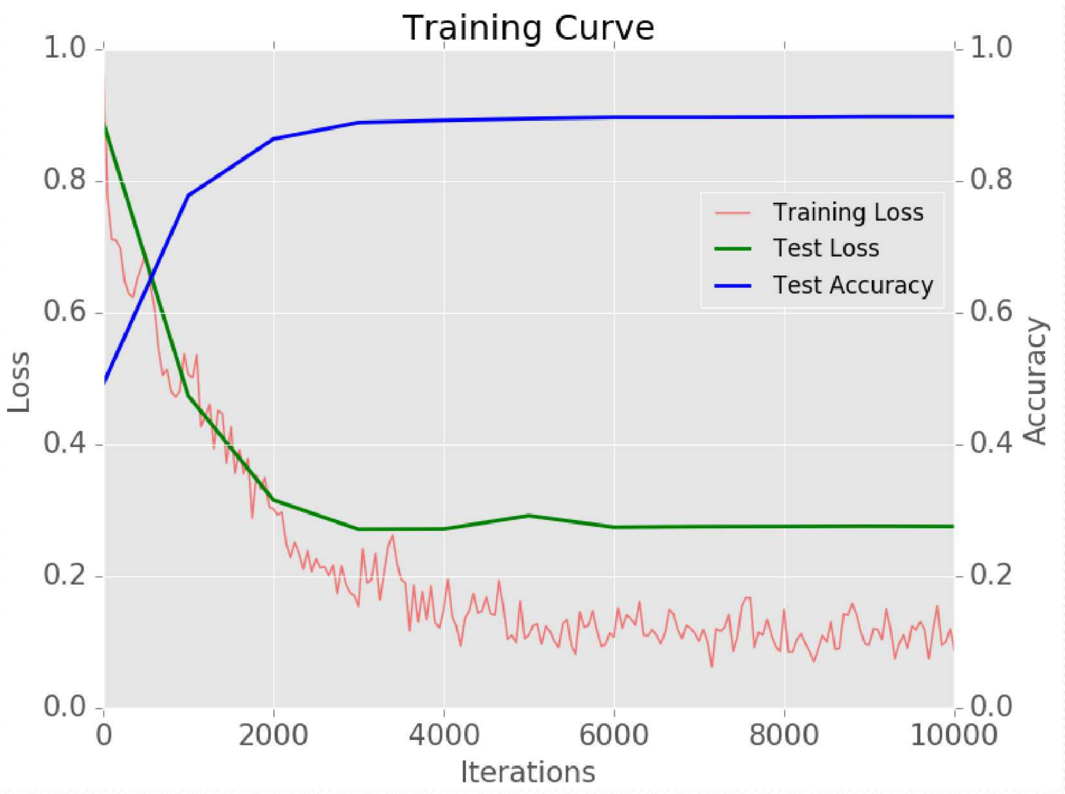






Stages of Deep Learning

- Collect Data
  - Understand
  - Filter
  - Normalize
    - Size
    - Magnitude
  - Augment
- Design DNN
  - Layers
  - Elements/Layer
  - Output
- Train
- Test/Evaluate
  - Confusion Matrix
  - Misclassified samples
- REPEAT



Stages of Deep Learning

- Collect Data
  - Understand
  - Filter
  - Normalize
    - Size
    - Magnitude
  - Augment
- Design DNN
  - Layers
  - Elements/Layer
  - Output
- Train
- Test/Evaluate
  - Confusion Matrix
  - Misclassified samples
- REPEAT

Confusion Matrix

	Eng	Spa	Dar	Fre	Pas	Rus	Urd	Chi
Eng	247	14	10	7	1	24	25	49
Spa	8	333	3	7	5	11	8	10
Dar	10	33	176	24	58	25	39	23
Fre	24	16	2	274	6	32	9	32
Pas	10	15	33	9	225	37	37	29
Rus	3	7	0	5	1	222	7	11
Urd	6	19	4	5	24	11	263	15
Chi	11	7	3	5	3	16	8	346

DL Software

- Many frameworks available for application development
  - Code supporting papers
  - Use of pretrained models

Software	License	Open	Platform	Written in	Interface
roNNie.ai	MIT	Yes	Linux, macOS, Windows	Python	Python
BigDL	Apache 2.0	Yes	Apache Spark	Scala	Scala, Python
Caffe	BSD	Yes	Linux, macOS, Windows	C++	Python, MATLAB, C++
Deeplearning4j	Apache 2.0	Yes	Linux, macOS, Windows, Android	C++, Java	Java, Scala, Clojure, Python (Keras), Kotlin
Chainer	MIT	Yes	Linux, macOS, Windows		Python
Darknet	Public Domain	Yes	Cross-Platform	C	C, Python
Dlib	Boost	Yes	Cross-Platform	C++	C++
DataMelt (DMelt)	Freemium	Yes	Cross-Platform	Java	Java
DyNet	Apache 2.0	Yes	Linux, macOS, Windows		C++, Python
Intel Data Analytics Acceleration Library	Apache 2.0	Yes	Linux, macOS, Windows on Intel CPU	C++, Python, Java	C++, Python, Java
Intel Math Kernel Library	Proprietary	No	Linux, macOS, Windows on Intel CPU		C
Keras	MIT	Yes	Linux, macOS, Windows	Python	Python, R
MATLAB + NN Toolbox	Proprietary	No	Linux, macOS, Windows	C, C++, Java, MATLAB	MATLAB
Microsoft Cognitive Toolkit	MIT	Yes	Windows, Linux, macOS on roadmap	C++	Python, C++, Command line, BrainScript (.NET on roadmap)
Apache MXNet	Apache 2.0	Yes	Linux, macOS, Windows, AWS, Android, iOS, JavaScript	Small C++ core library	C++, Python, Julia, Matlab, JavaScript, Go, R, Scala, Perl
Neural Designer	Proprietary	No	Linux, macOS, Windows	C++	Graphical user interface
OpenNN	GNU LGPL	Yes	Cross-platform	C++	C++
PlaidML	AGPL3	Yes	Linux, macOS, Windows	C++, Python	Keras, Python, C++, C
PyTorch	BSD	Yes	Linux, macOS, Windows	Python, C, CUDA	Python
Apache SINGA	Apache 2.0	Yes	Linux, macOS, Windows	C++	Python, C++, Java
TensorFlow	Apache 2.0	Yes	Linux, macOS, Windows, Android	C++, Python, CUDA	Python, C/C++, Java, Go, R, Julia
TensorLayer	Apache 2.0	Yes	Linux, macOS, Windows, Android	C++, Python,	Python
Theano	BSD	Yes	Cross-platform	Python	Python
Torch	BSD	Yes	Linux, macOS, Windows, Android, iOS	C, Lua	Lua, LuaJIT, C, C++/OpenCL
Wolfram Mathematica	Proprietary	No	Windows, macOS, Linux, Cloud	C++, Wolfram, CUDA	Wolfram Language
VerAI	Proprietary	No	Linux, Web-based	C++,Python, Go, Angular	Graphical user interface, cli

https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

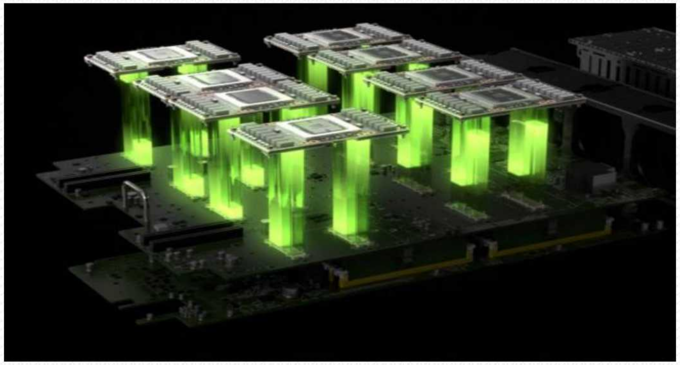
Hardware

- Embarrassingly Parallel Matrix-Vector Multiplications
- CPU
- GPU
  - Boards
  - Clusters
- Supercomputers

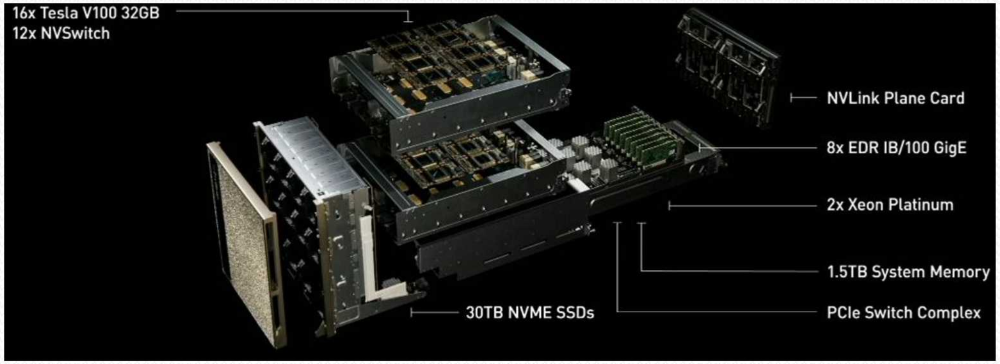
Sandia DL Hardware

Training

6 x DGX-1



4 x DGX-2

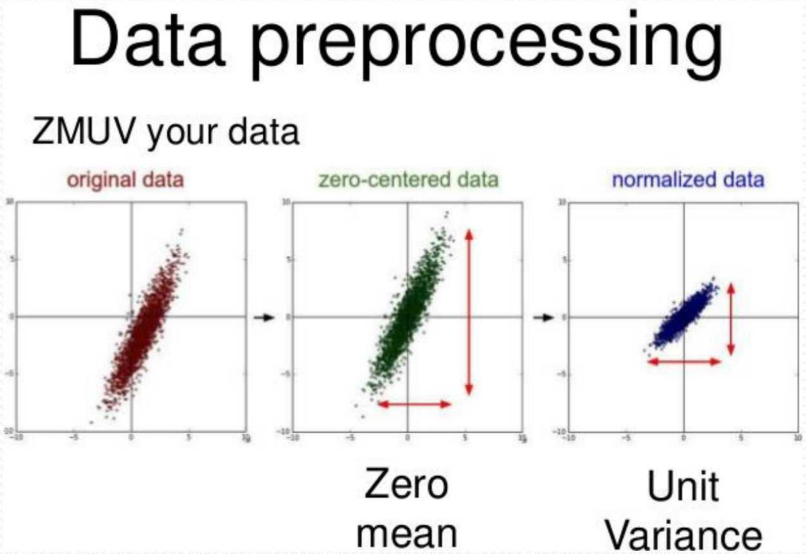


Inference Analytics

20 x PC40 PCIe Nvidia Pascal Systems

Data

- Deep Learning is data-driven
- Need for data engineering
  - Pre-processing
    - Whitening
    - Filtering
    - Log-scaling
  - Labels
    - Class
    - Real-valued number (e.g., probability)
  - Data augmentation
  - Minibatches
  - Class distributions



Data Augmentation

- Minor adjustments
  - Adding noise
  - Image rotation, scaling, translation, cropping, occlusion
- Simulation

Sasquatch DeepCloud Demo