Deep Learning

Deep Learning in **One** Slide

- **What is it:**
  Extract useful patterns from data.
- **How:**
  Neural network + optimization
- **How (Practical):**
  Python + TensorFlow & friends
- **Hard Part:**
  Good Questions + Good Data
- **Why now:**
  Data, hardware, community, tools, investment
- **Where do we stand?**
  Most big questions of intelligence have not been answered nor properly formulated

Exciting progress:

- Face recognition
- Image classification
- Speech recognition
- Text-to-speech generation
- Handwriting transcription
- Machine translation
- Medical diagnosis
- Cars: drivable area, lane keeping
- Digital assistants
- Ads, search, social recommendations
- Game playing with deep RL

History of Deep Learning Ideas and Milestones*

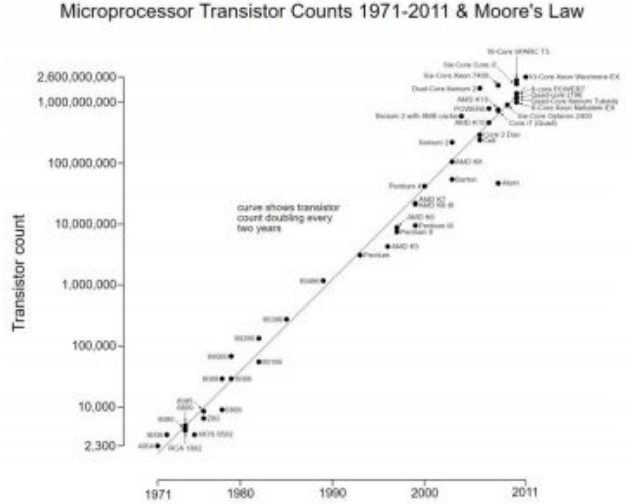
Perspective:

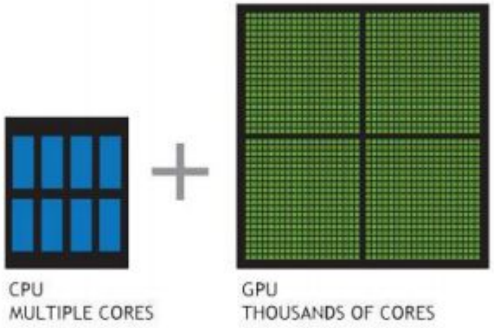
- Universe created
  13.8 billion years ago
- Earth created
  4.54 billion years ago
- Modern humans
  300,000 years ago
- Civilization
  12,000 years ago
- Written record
  5,000 years ago

- 1943: Neural networks
- 1957: Perceptron
- 1974-86: Backpropagation, RBM, RNN
- 1989-98: CNN, MNIST, LSTM, Bidirectional RNN
- 2006: "Deep Learning", DBN
- 2009: ImageNet
- 2012: AlexNet, Dropout
- 2014: GANs
- 2014: DeepFace
- 2016: AlphaGo
- 2017: AlphaZero, Capsule Networks
- 2018: BERT

* Dates are for perspective and not as definitive historical record of invention or credit

Deep Learning Breakthroughs: What Changed?





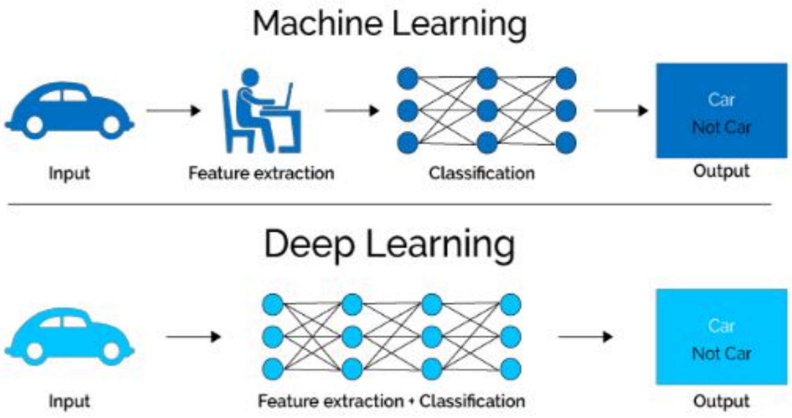
- **Compute**
  CPUs, GPUs, ASICs
- **Organized large(-ish) datasets**
  Imagenet
- **Algorithms and research:**
  Backprop, CNN, LSTM
- **Software and Infrastructure**
  Git, ROS, PR2, AWS, Amazon Mechanical Turk, TensorFlow, …
- **Financial backing of large companies**
  Google, Facebook, Amazon, …

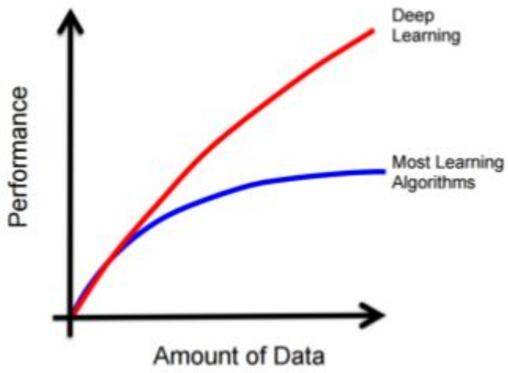
The Challenge of Deep Learning:

Efficient Teaching + Efficient Learning

- Humans can learn from very few examples
- Machines (in most cases) need thousands/millions of examples

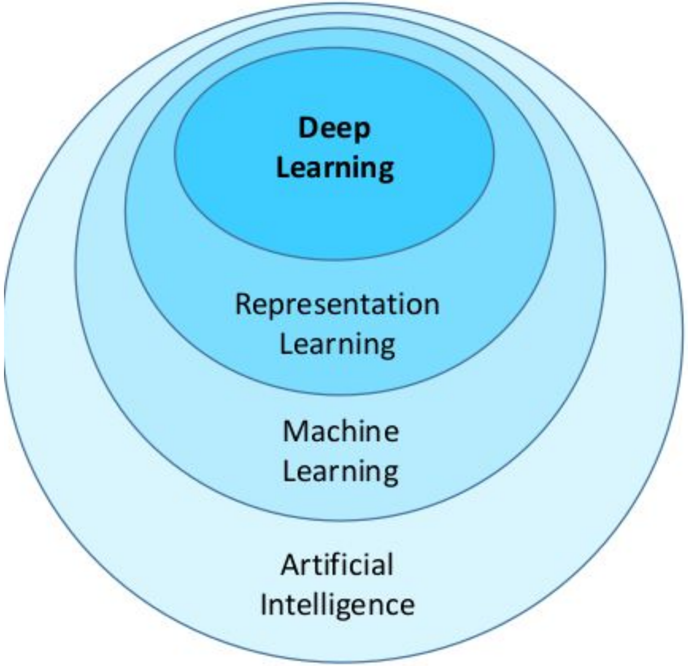
Why Deep Learning? **Scalable** Machine Learning

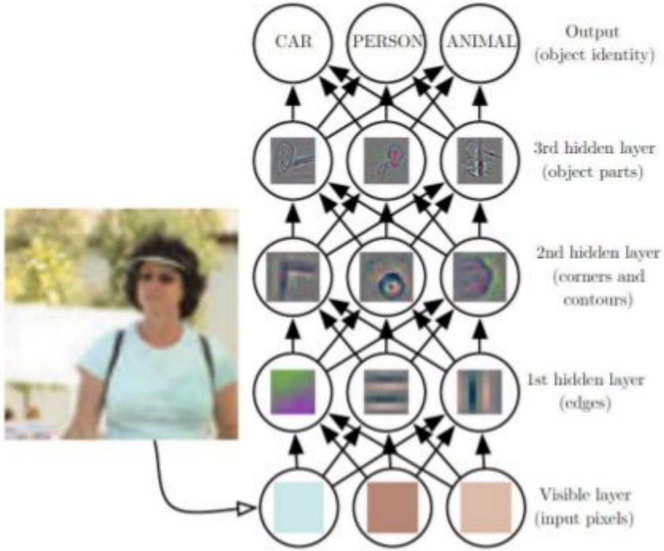




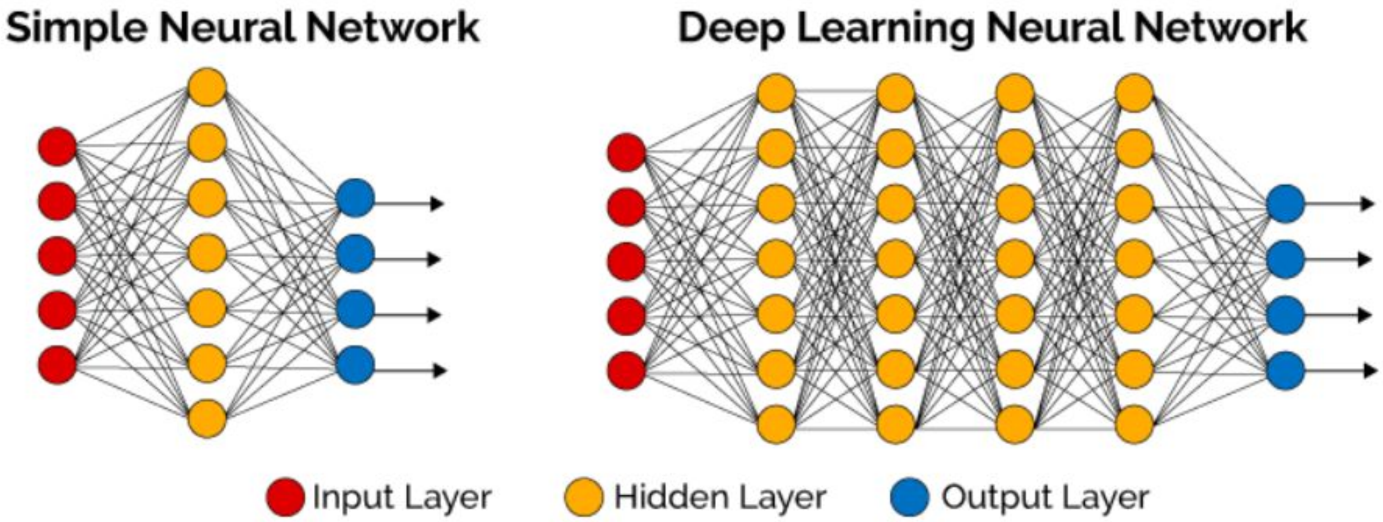
Deep Learning is **Representation Learning**

(aka Feature Learning)



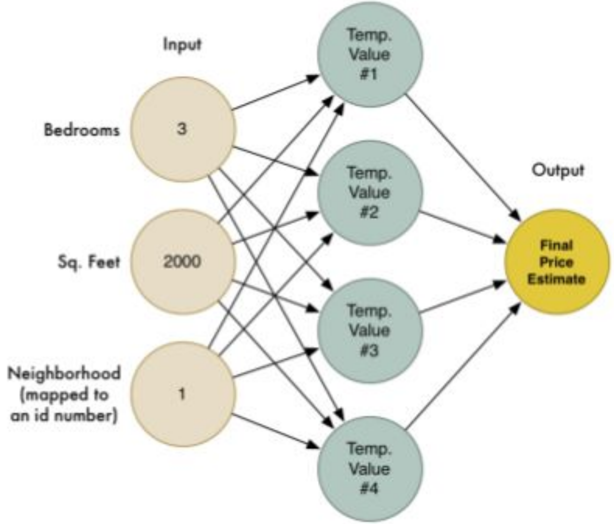


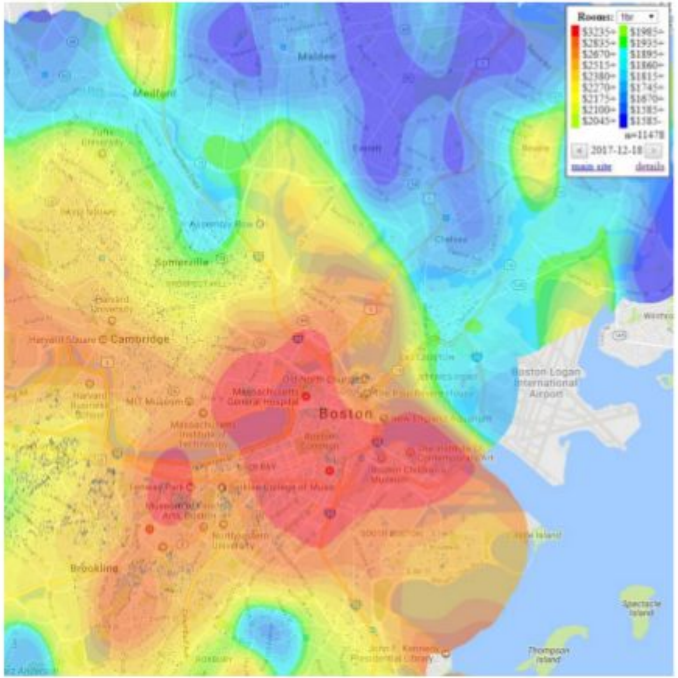
Combing Neurons in Hidden Layers:
The "Emergent" Power to Approximate



Universality: For any arbitrary function f(x), there exists a neural network that closely approximate it for any input x

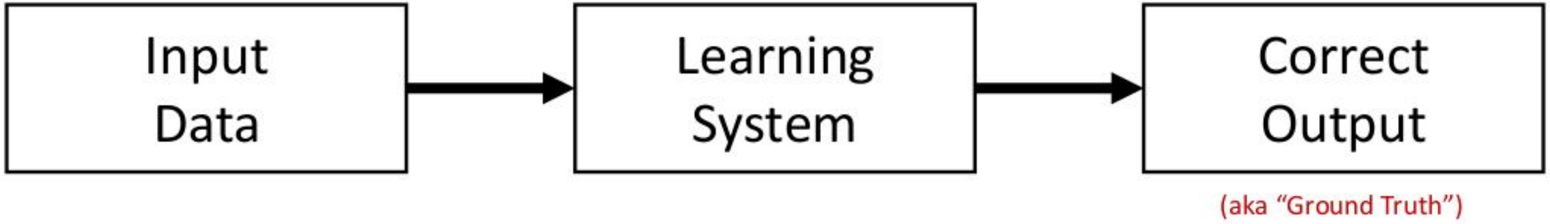
Special Purpose Intelligence:
Estimating Apartment Cost



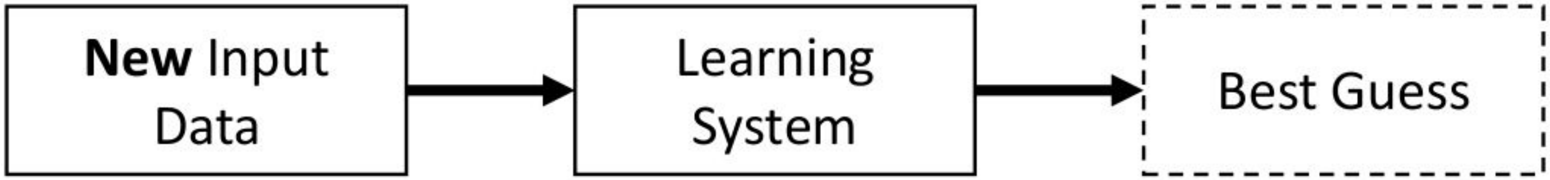


Deep Learning: Training and Testing

Training Stage:



Testing Stage:



How Neural Networks Learn: Backpropagation

Forward Pass:

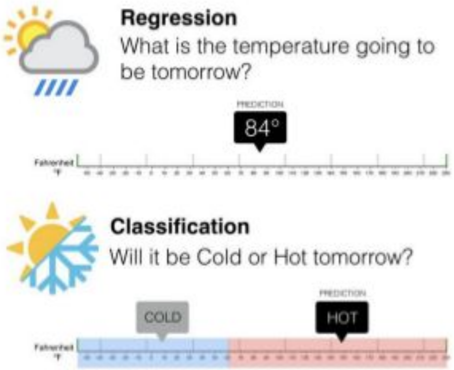
Input Data → Neural Network → Prediction

Backward Pass (aka Backpropagation):

Neural Network ← Measure of Error

Adjust to Reduce Error

Loss Functions



- Loss function quantifies gap between prediction and ground truth
- For regression:
  - Mean Squared Error (MSE)
- For classification:
  - Cross Entropy Loss

Mean Squared Error

Prediction

$$MSE = \frac{1}{N} \sum (t_i - s_i)^2$$

Ground Truth

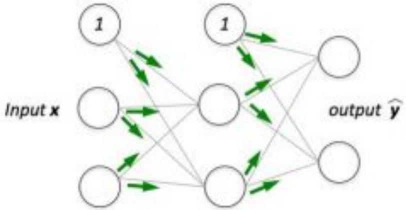
Cross Entropy Loss

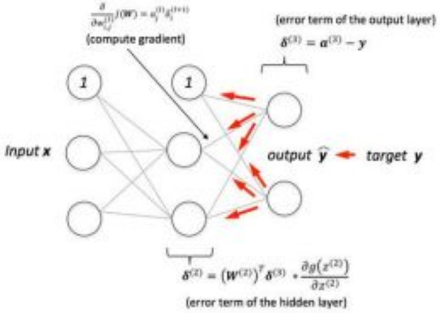
Classes Prediction

$$CE = -\sum_{i}^{C} t_i log(s_i)$$

Ground Truth {0,1}

Key Concepts:
Backpropagation





Task: Update the **weights** and **biases** to decrease **loss function**

Subtasks:

1. Forward pass to compute network output and "error"
2. Backward pass to compute gradients
3. A fraction of the weight's gradient is subtracted from the weight.

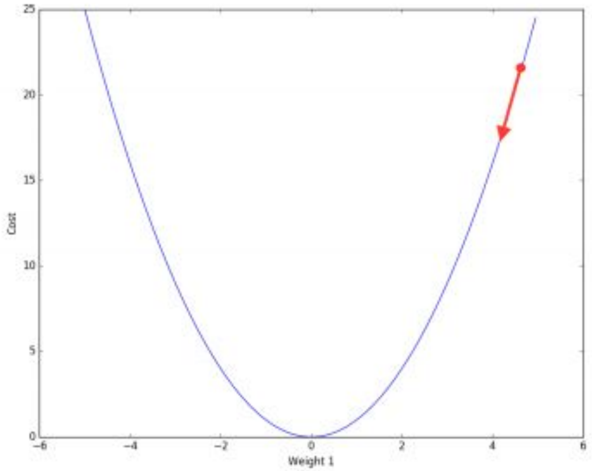
Learning Rate

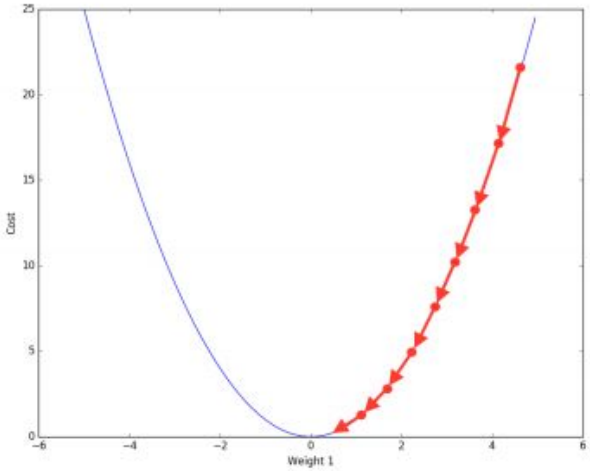
Loss function:

$$C = \frac{(y-a)^2}{2}$$

Learning is an Optimization Problem

Task: Update the **weights** and **biases** to decrease **loss function**





Use mini-batch or stochastic gradient descent.

Mini-Batch Size

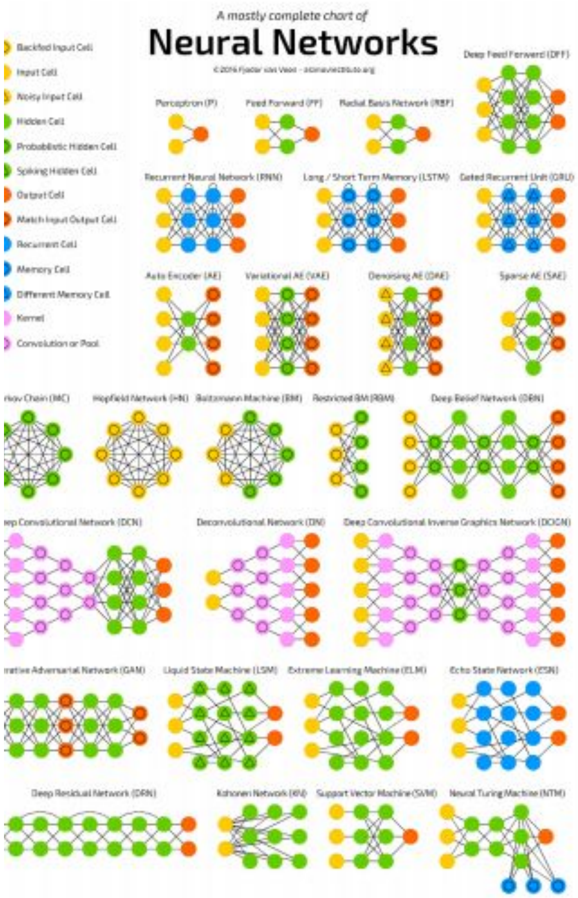
Mini-Batch size: Number of training instances the network evaluates per weight update step.

- Larger batch size = more computational speed
- Smaller batch size = (empirically) better generalization

"Training with large minibatches is bad for your health. More importantly, it's bad for your test error. Friends don't let friends use minibatches larger than 32."
- Yann LeCun

Revisiting Small Batch Training for Deep Neural Networks (2018)

Useful Deep Learning Terms



- Basic terms:
  - **Deep Learning ≈ Neural Networks**
  - **Deep Learning** is a subset of **Machine Learning**
- Terms for neural networks:
  - **MLP:** Multilayer Perceptron
  - **DNN:** Deep neural networks
  - **RNN:** Recurrent neural networks
    - **LSTM:** Long Short-Term Memory
  - **CNN:** Convolutional neural networks
  - **DBN:** Deep Belief Networks
- Neural network operations:
  - Convolution
  - Pooling
  - Activation function
  - Backpropagation

Computer Vision

Deep Learning is Hard:

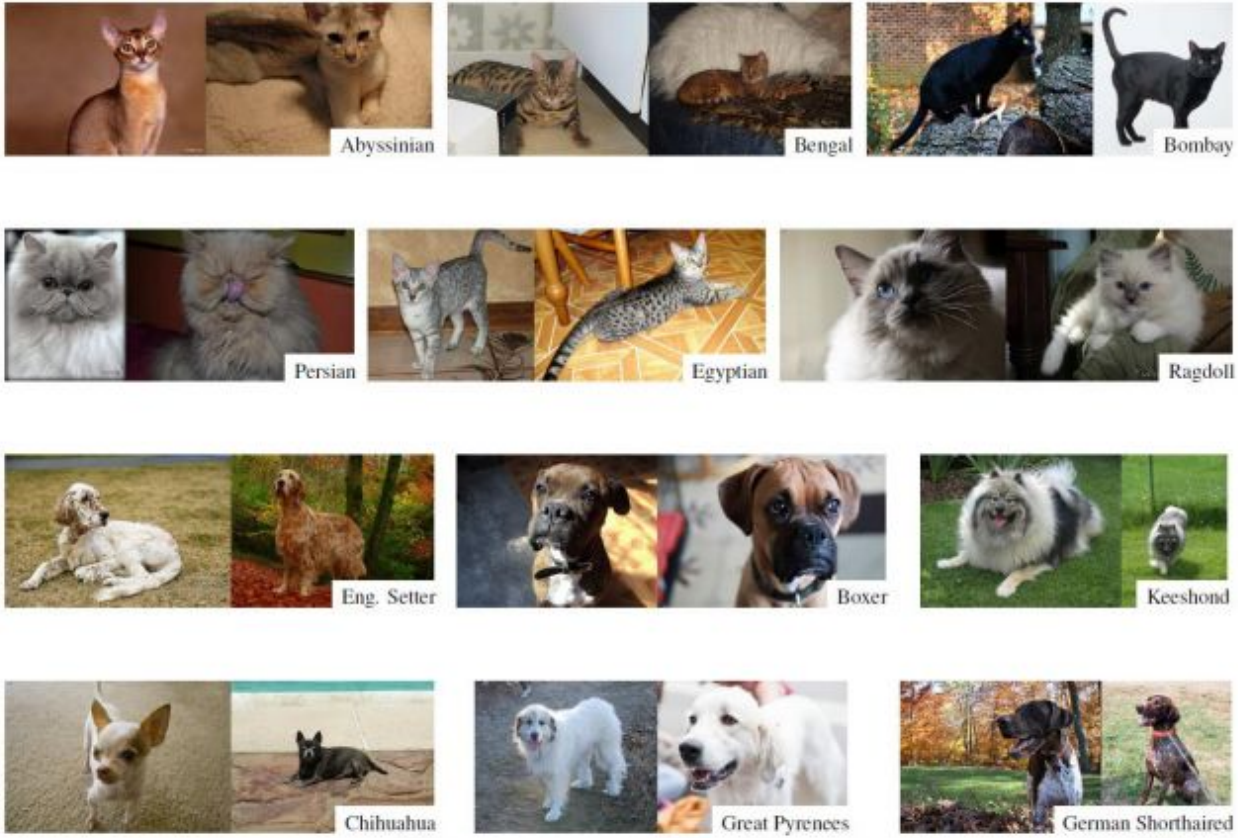
Illumination Variability

Deep Learning is Hard:

Pose Variability and Occlusions

gure 1. **The deformable and truncated cat.** Cats exhibit (

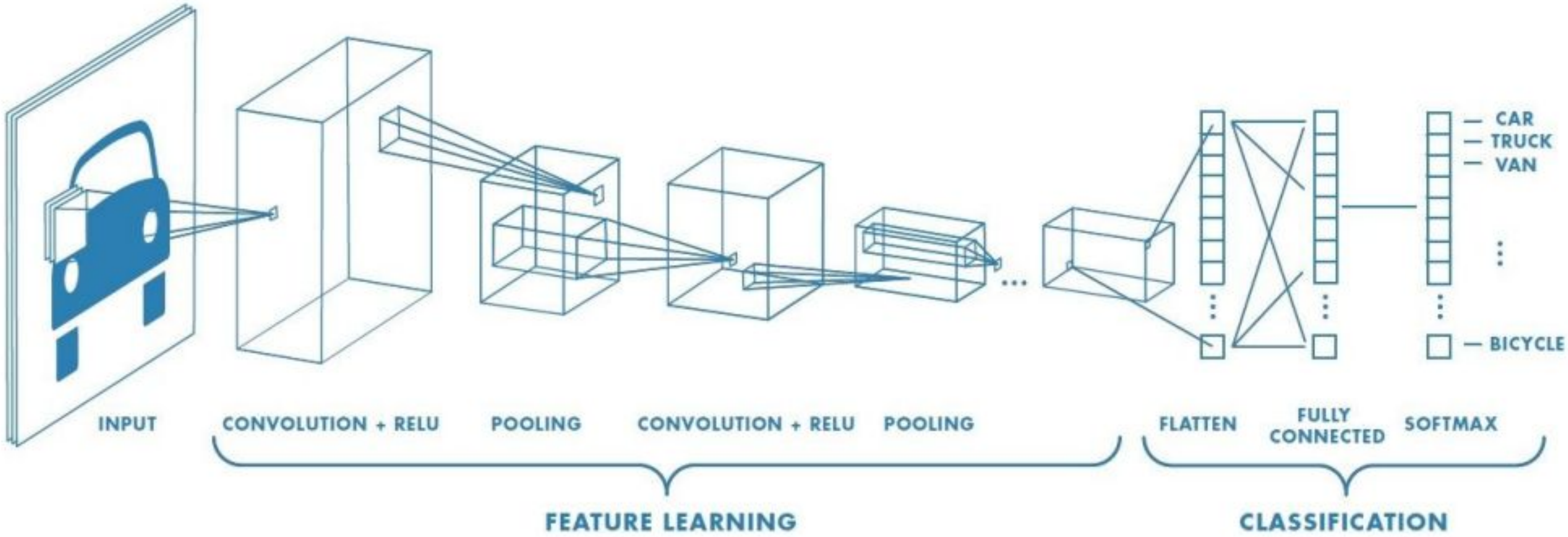
Deep Learning is Hard:
Intra-Class Variability



Parkhi et al. "Cats and dogs." 2012.

Convolutional Neural Networks (CNN)

Sample Architecture



What is ImageNet?

- **ImageNet:** dataset of 14+ million images (21,841 categories)
- Let's take the high level category of **fruit** as an example:
  - Total 188,000 images of fruit
  - There are 1206 Granny Smith apples:

Human error (5.1%) surpassed in 2015

- **AlexNet (2012): First CNN (15.4%)**
  - 8 layers
  - 61 million parameters
- **ZFNet (2013): 15.4% to 11.2%**
  - 8 layers
  - More filters. Denser stride.
- **VGGNet (2014): 11.2% to 7.3%**
  - Beautifully uniform:
    3x3 conv, stride 1, pad 1, 2x2 max pool
  - 16 layers
  - 138 million parameters
- **GoogLeNet (2014): 11.2% to 6.7%**
  - Inception modules
  - 22 layers
  - 5 million parameters
    (throw away fully connected layers)
- **ResNet (2015): 6.7% to 3.57%**
  - More layers = better performance
  - 152 layers
- **CUImage (2016): 3.57% to 2.99%**
  - Ensemble of 6 models
- **SENet (2017): 2.99% to 2.251%**
  - Squeeze and excitation block: network is allowed to adaptively adjust the weighting of each feature map in the convolutional block.

How ?

Data Augmentation

Crop:

Flip:

Scale:

Rotate:

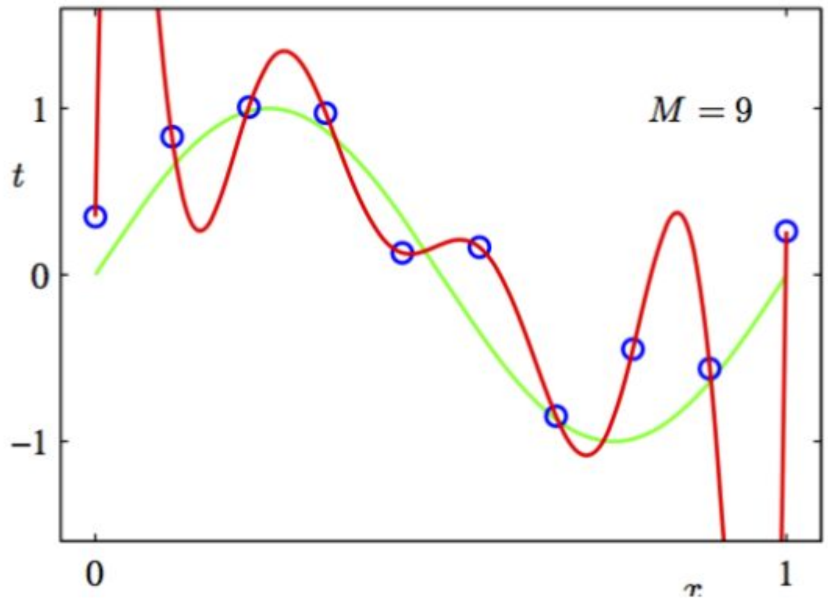
Translation:

Noise:

Key Concepts:

Overfitting and Regularization

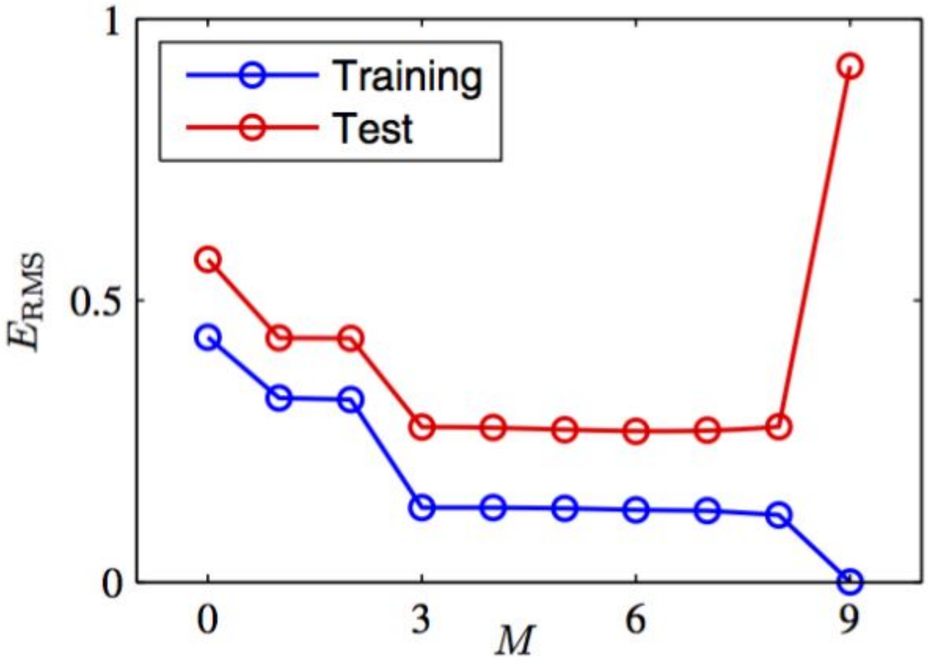
- Help the network **generalize** to data it hasn't seen.
- Big problem for **small datasets**.
- Overfitting example (a sine curve vs 9-degree polynomial):



Key Concepts:

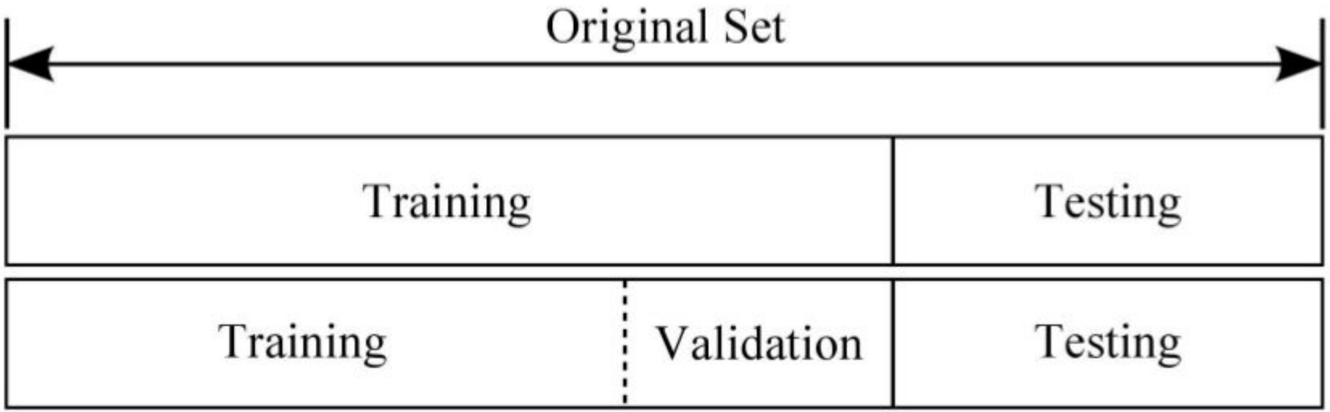
Overfitting and Regularization

- Overfitting: The error decreases in the training set but increases in the test set.



Key Concepts:

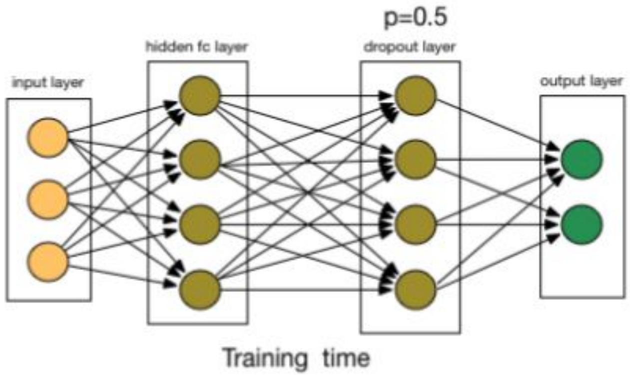
Regularization: Early Stoppage



- Create “validation” set (subset of the training set).
  - Validation set is assumed to be a representative of the testing set.
- **Early stoppage:** Stop training (or at least save a checkpoint) when performance on the validation set decreases

Key Concepts:

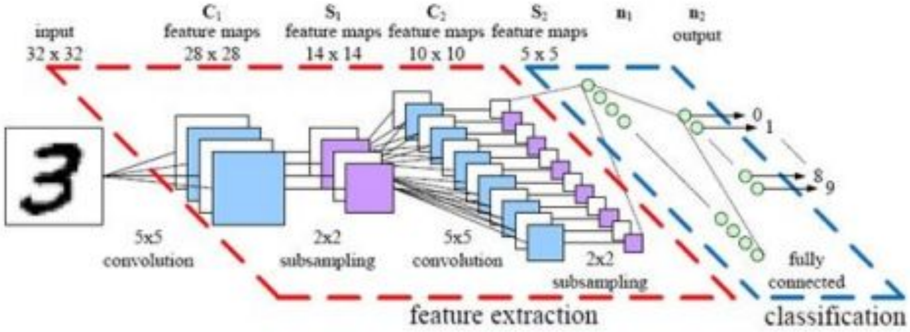
Regularization: Dropout



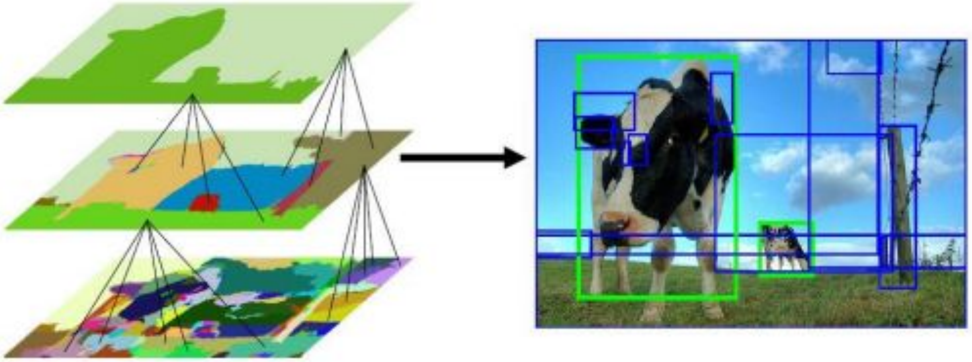
- **Dropout:** Randomly remove some nodes in the network (along with incoming and outgoing edges)
- Notes:
  - Usually *p >= 0.5* (*p* is probability of keeping node)
  - Input layers *p* should be much higher (and use noise instead of dropout)
  - Most deep learning frameworks come with a dropout layer

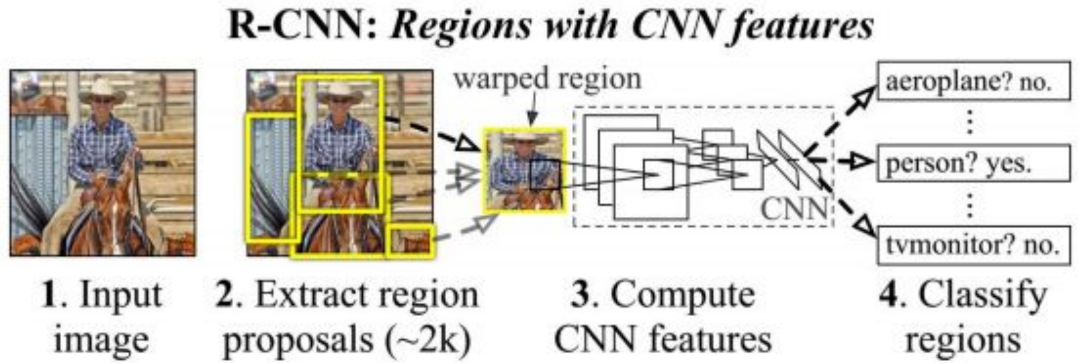
CV : Applications

Object Recognition / Classification

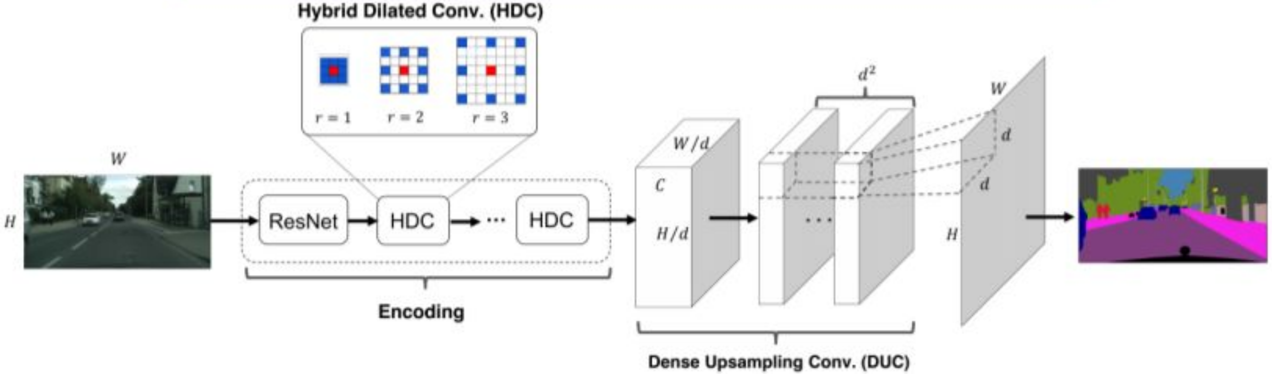


Object Detection

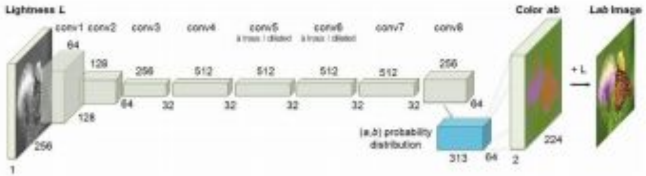




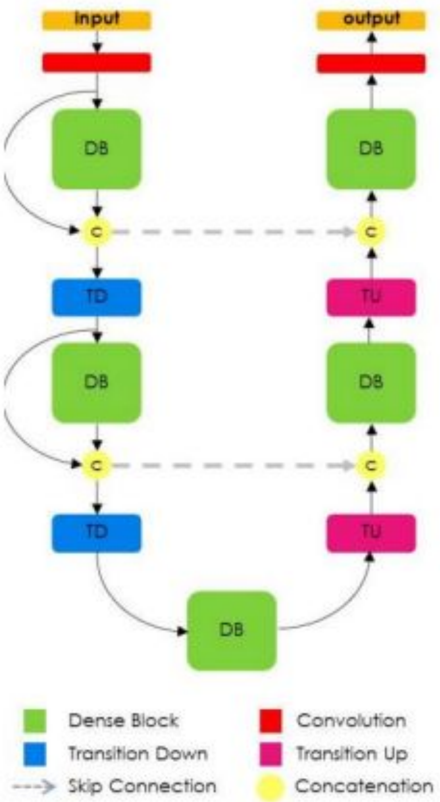
Semantic Segmentation



Colorization of Images



Background Removal (2017)



Applications: Image Caption Generation

a man sitting on a couch with a dog
a man sitting on a chair with a dog in his lap

Video Description Generation

Correct descriptions.

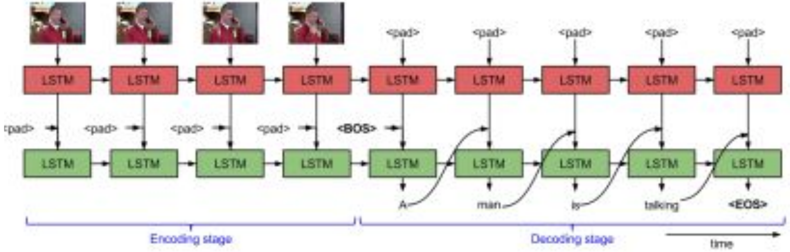
S2VT: A man is doing stunts on his bike.

S2VT: A herd of zebras are walking in a field.

Relevant but incorrect descriptions.

S2VT: A small bus is running into a building.

S2VT: A man is cutting a piece of a pair of a paper.

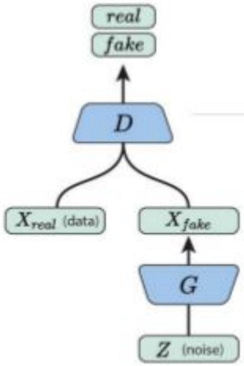


Venugopalan et al.
"Sequence to sequence-video to text." 2015.

Code: https://vsubhashini.github.io/s2vt.html

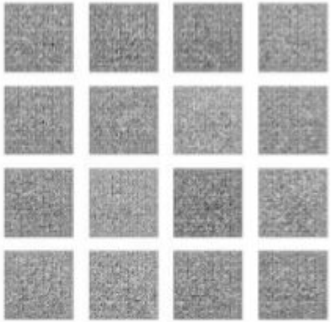
Generative Adversarial Network (GANs)

Generative Adversarial Networks (GANs) are a way to make a generative model by having two neural networks compete with each other.



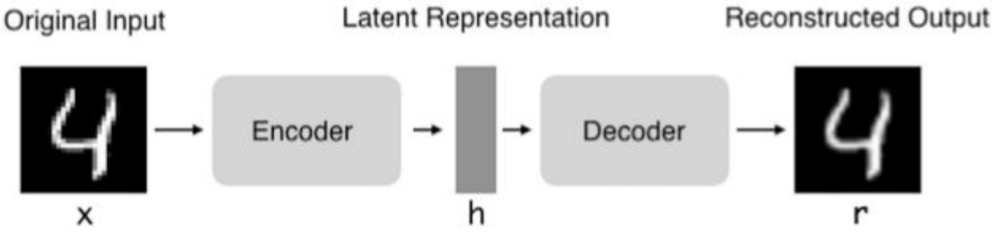
The **discriminator** tries to distinguish genuine data from forgeries created by the generator.

The **generator** turns random noise into immitations of the data, in an attempt to fool the discriminator.

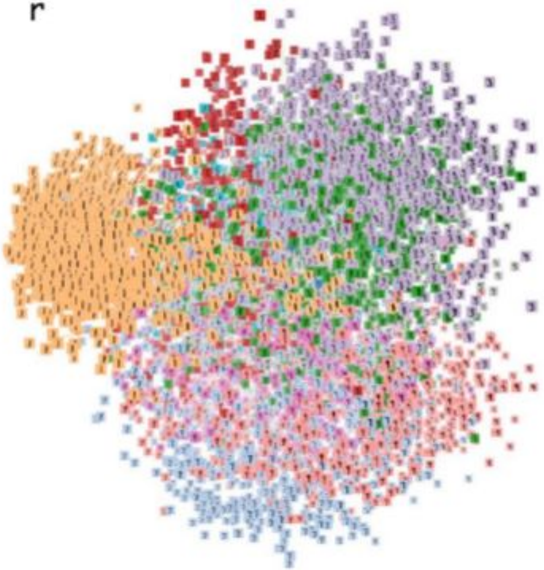


Progressive GAN
10/2017
1024 x 1024

Autoencoders



- Unsupervised learning
- Gives embedding
    - Typically better embeddings come from discriminative task



http://projector.tensorflow.org/

Current Challenges

- **Transfer learning:** Unable to transfer representation to most reasonably related domains except in specialized formulations.
  - **Understanding:** Lacks "reasoning" or ability to truly derive "understanding" as previously defined on anything but specialized problem formulations. (Definition used: Ability to turn complex information to into simple, useful information.)
- Requires **big** data: inefficient at learning from data
- Requires **supervised** data: costly to annotate real-world data
- **Not fully automated:** Needs hyperparameter tuning for training: learning rate, loss function, mini-batch size, training iterations, momentum, optimizer selection, etc.
- **Reward:** Defining a good reward function is difficult.
- **Transparency:** Neural networks are for the most part black boxes (for real-world applications) even with tools that visualize various aspects of their operation.
- **Edge cases:** Deep learning is not good at dealing with edge cases.

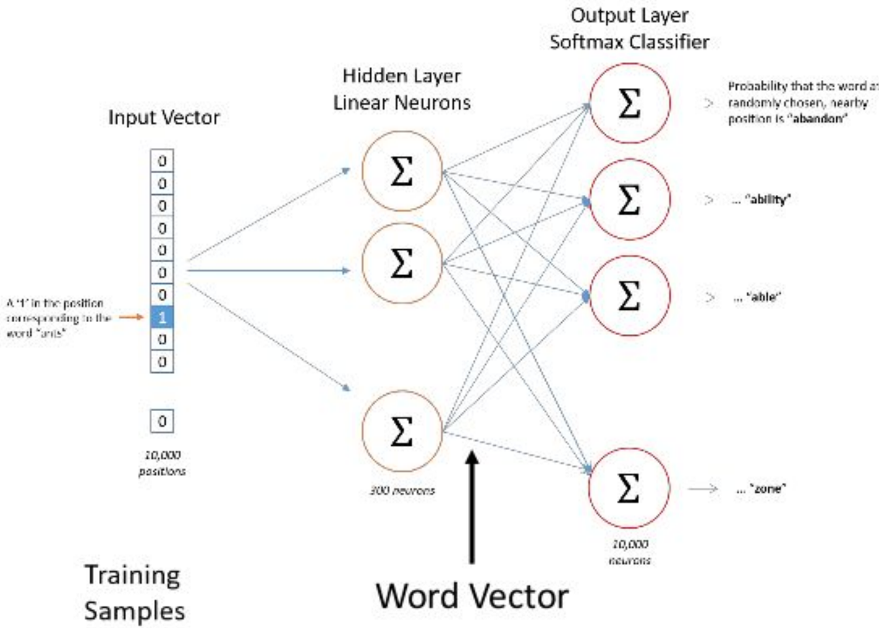
Robustness:

Fooled by a Little Distortion

correct +distort ostrich correct +distort ostrich

Natural Language Processing (NLP)

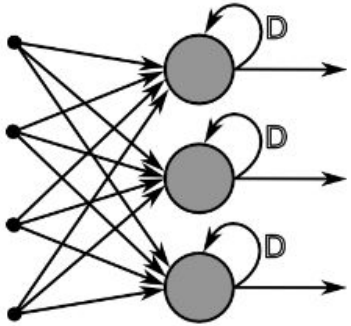
Word Embeddings (Word2Vec)

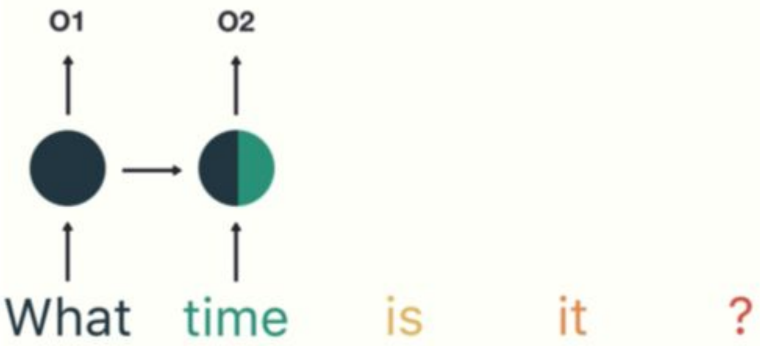


Skip Gram Model:

Source Text	Training Samples
The quick brown fox jumps over the lazy dog.	(the, quick) (the, brown)
The **quick** brown fox jumps over the lazy dog.	(quick, the) (quick, brown) (quick, fox)
The quick **brown** fox jumps over the lazy dog.	(brown, the) (brown, quick) (brown, fox) (brown, jumps)
The quick brown **fox** jumps over the lazy dog.	(fox, quick) (fox, brown) (fox, jumps) (fox, over)

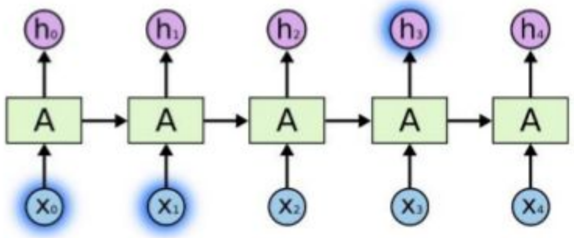
Recurrent Neural Networks





- Applications
  - Sequence Data
  - Text
  - Speech
  - Audio
  - Video
  - Generation

Long-Term Dependency



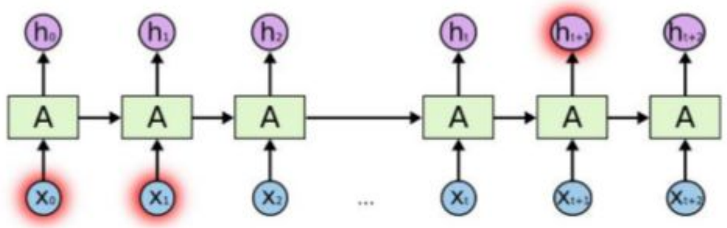
- Short-term dependence:

  Bob is eating an **apple.**

Context →

- Long-term dependence:

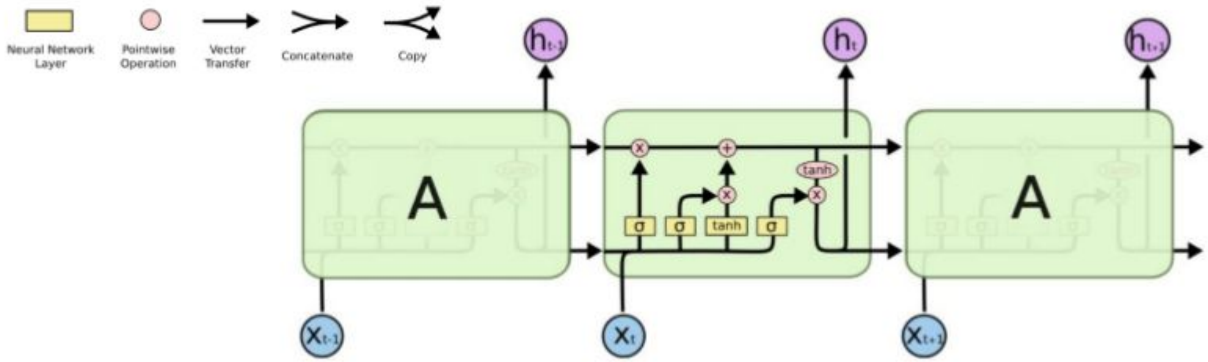
  Bob likes **apples**. He is hungry and decided to have a snack. So now he is eating an **apple.**



In theory, vanilla RNNs can handle arbitrarily long-term dependence.

In practice, it's difficult.

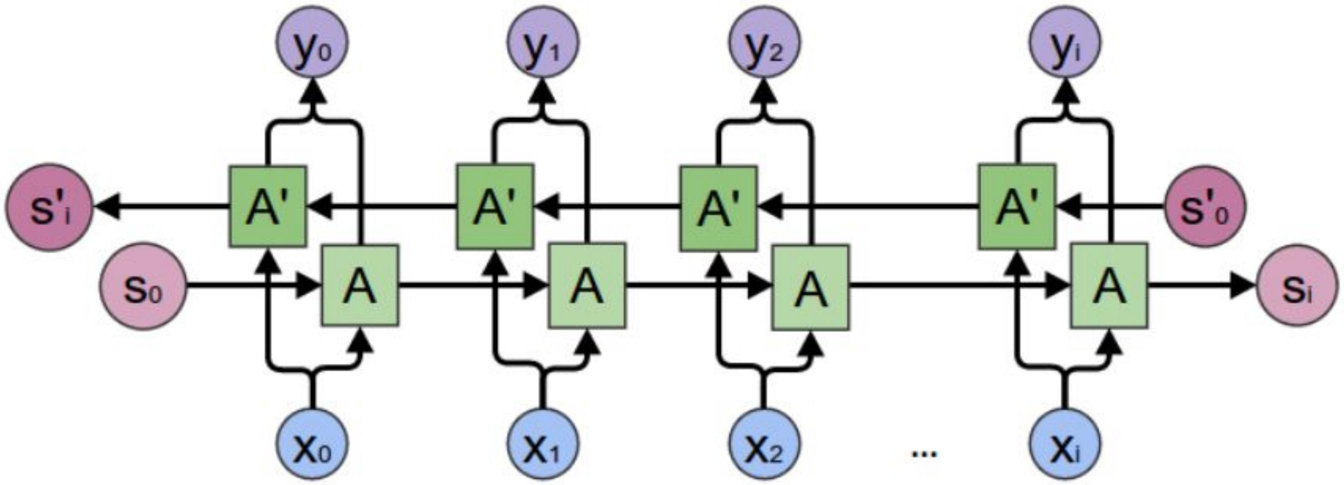
Long Short-Term Memory (LSTM) Networks:
Pick What to Forget and What To Remember



Conveyer belt for **previous state** and **new data**:

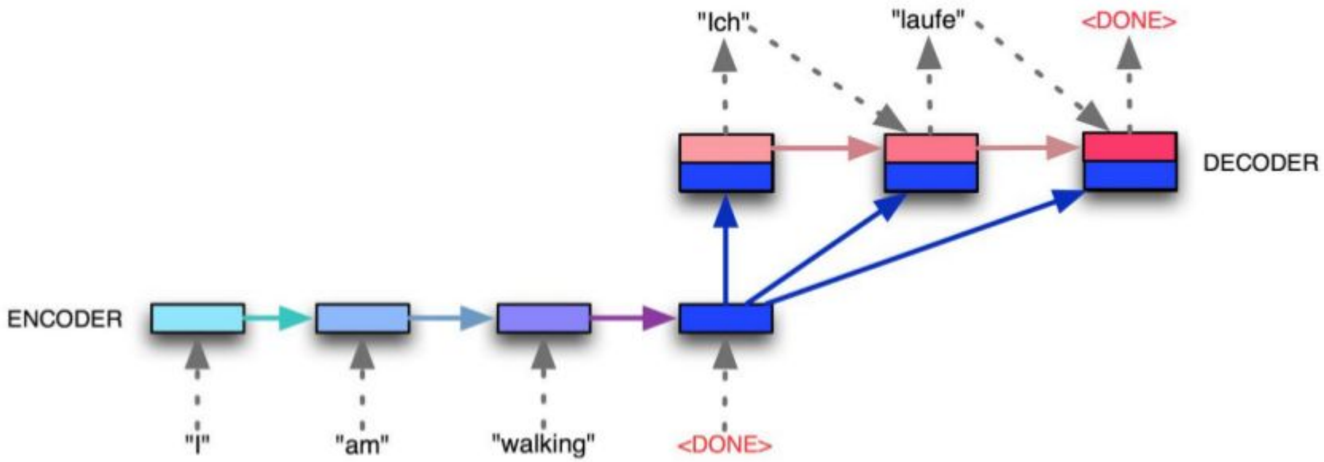
1. Decide what to forget (state)
2. Decide what to remember (state)
3. Decide what to output (if anything)

Bidirectional RNN



- Learn representations from both previous time steps and future time steps

Encoder-Decoder Architecture



Encoder RNN encodes input sequence into a fixed size vector, and then is passed repeatedly to decoder RNN.

NLP : Applications

More Deeper Application of NLP

Group 1	Group 2	Group 3
Cleanup, Tokenization	Information Retrieval and Extraction (IR)	Machine Translation
Stemming	Relationship Extraction	Automatic Summarization/ Paraphracing
Lemmatization	Named Entity Recognation (NER)	Natural Language Generation
Part of Speech Tagging	Sentiment Analysis/Sentance Boundary Dismbiguation	Reasoning over Knowledge Based
Query Expansion	World sense and Dismbiguation	Quation Answering System
Parsing	Text Similarity	Dialog System
Topic Segmentationand Recognation	Coreference Resolution	Image Captioning & other Multimodel Tasks
Morphological Degmentation (Word/Sentences)	Discourse Analysis	