CORNELL TECH

Deep Learning Clinic
(DLC)

Lecture 3
A Brief Introduction to Deep Learning

Jin Sun

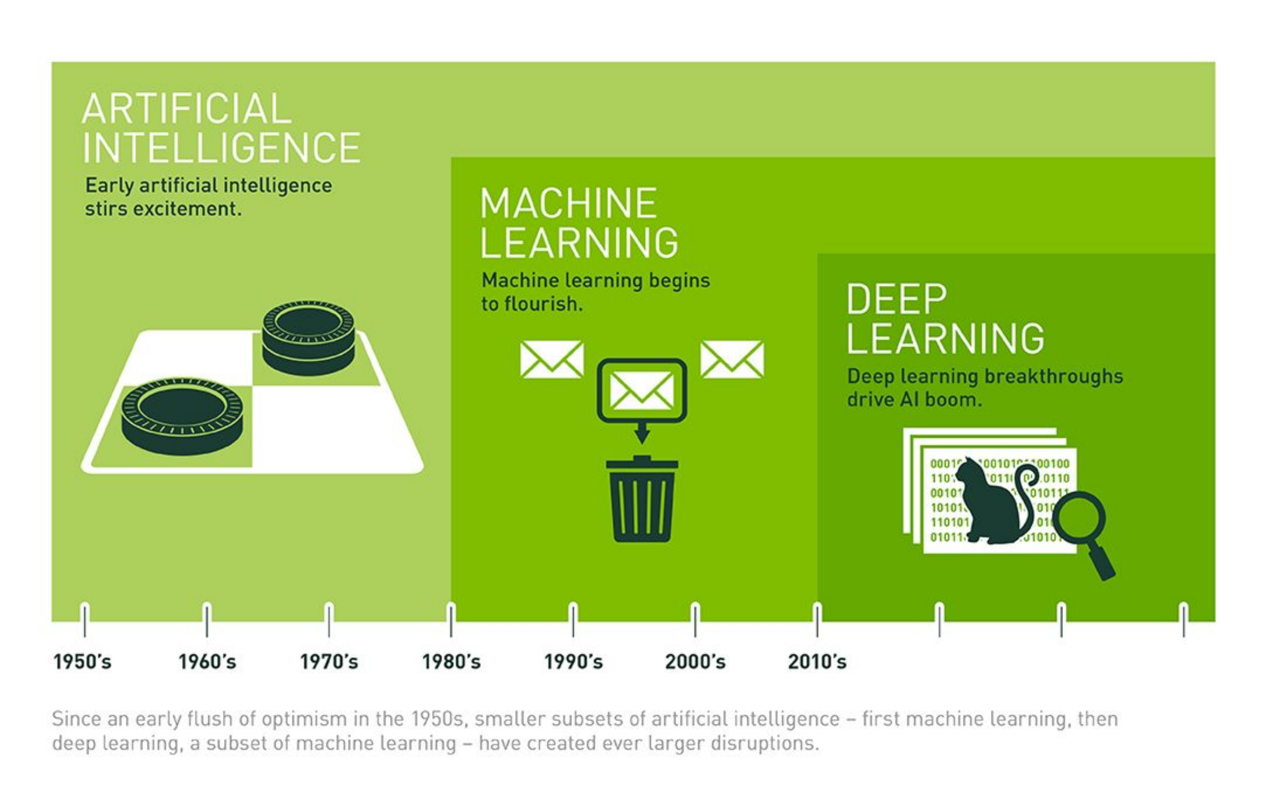
9/24/2019

Today

- **Overview**
- Basic Feedforward Networks and Core Concepts
- Deeper Networks: Challenges and Solutions
- Convolutional Networks
- Recurrent Networks
- Generative Models
- Research Frontiers

Some slides adapted from Ian Goodfellow.

Recall: relations with AI and ML

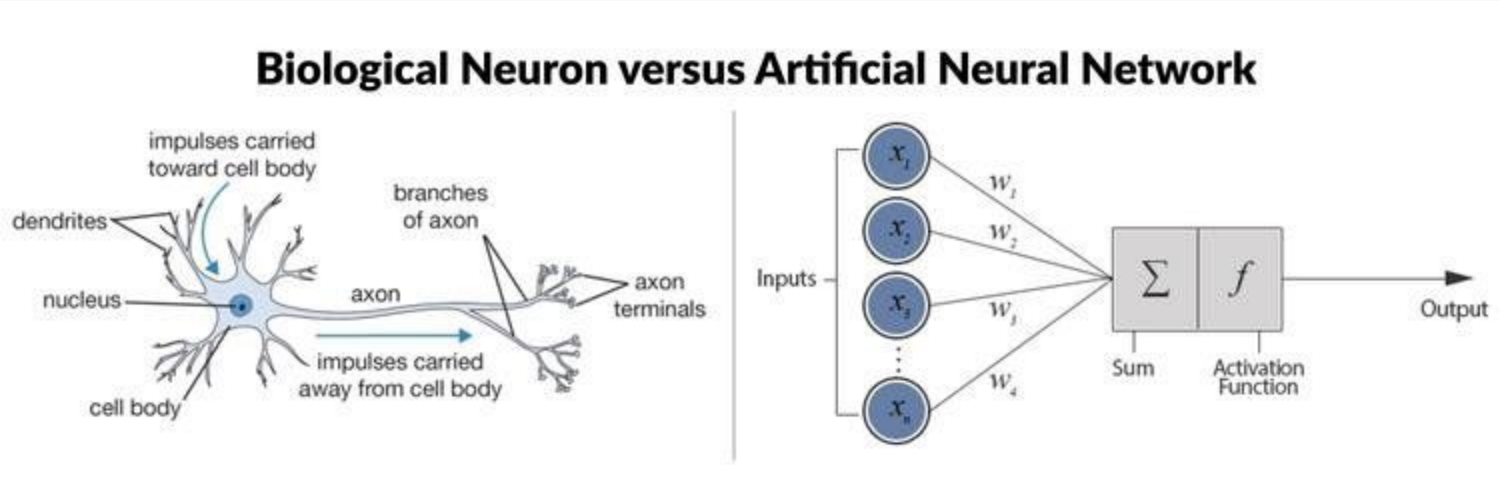


https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

Deep Learning is all about Neural Networks

(Artificial) Neural Network:

Originally inspired by biological neural networks.



https://www.researchgate.net/figure/Biological-Neuron-versus-Artificial-Neural-Network_fig4_325870973

Deep Learning is all about Neural Networks

Neural Network (Modern perspective):

A highly non-linear function that maps input to output.

Layered structure: each layer encodes different level of information.

Universal Approximation Theorem:

A single hidden layer feed-forward neural network can approximate almost **any** continuous function.

Neural Networks 90s vs Now

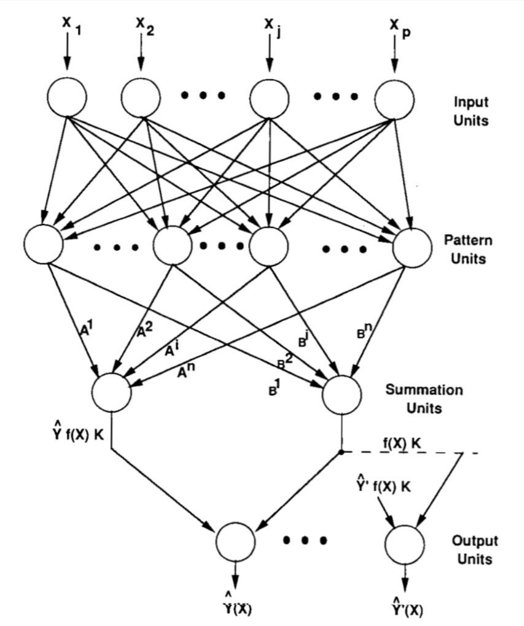
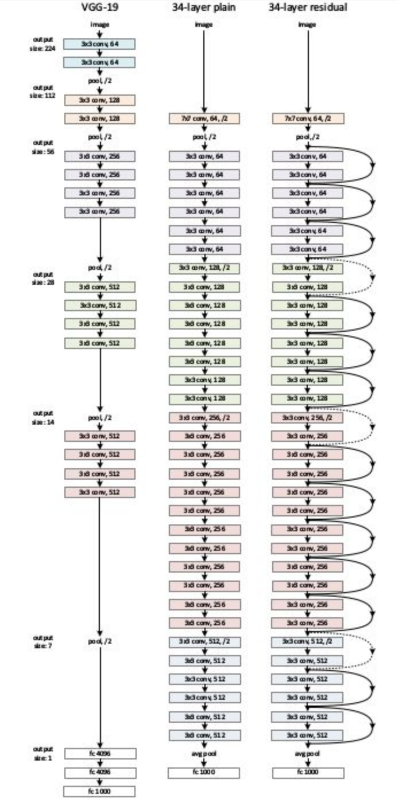


Fig. 1. GRNN block diagram.

1991

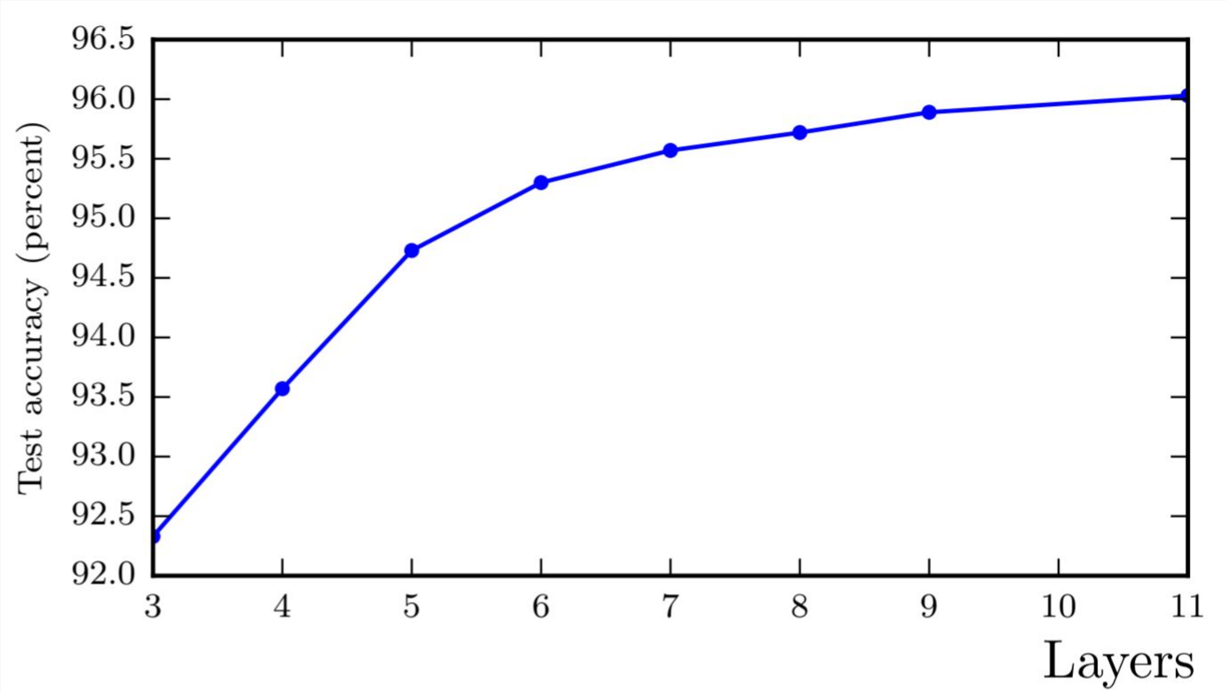
VS



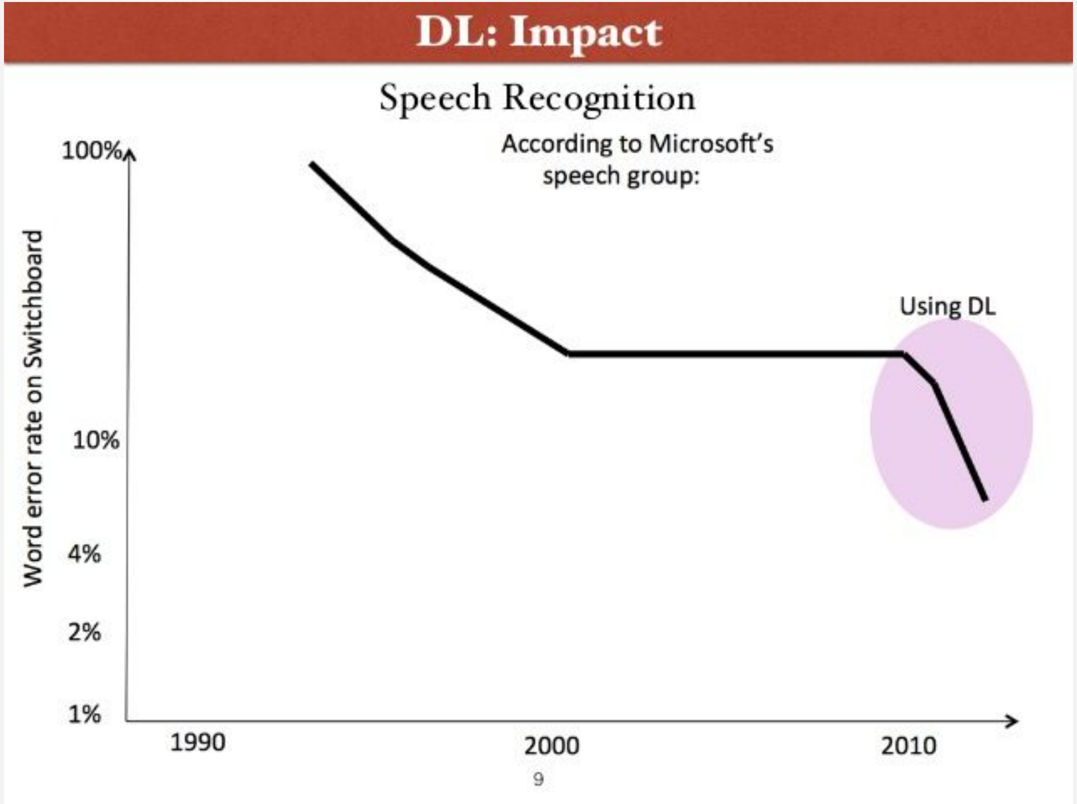
2016

Specht, Donald F. "A general regression neural network." IEEE transactions on neural networks 2.6 (1991): 568-576.

The Benefit of Going Deeper

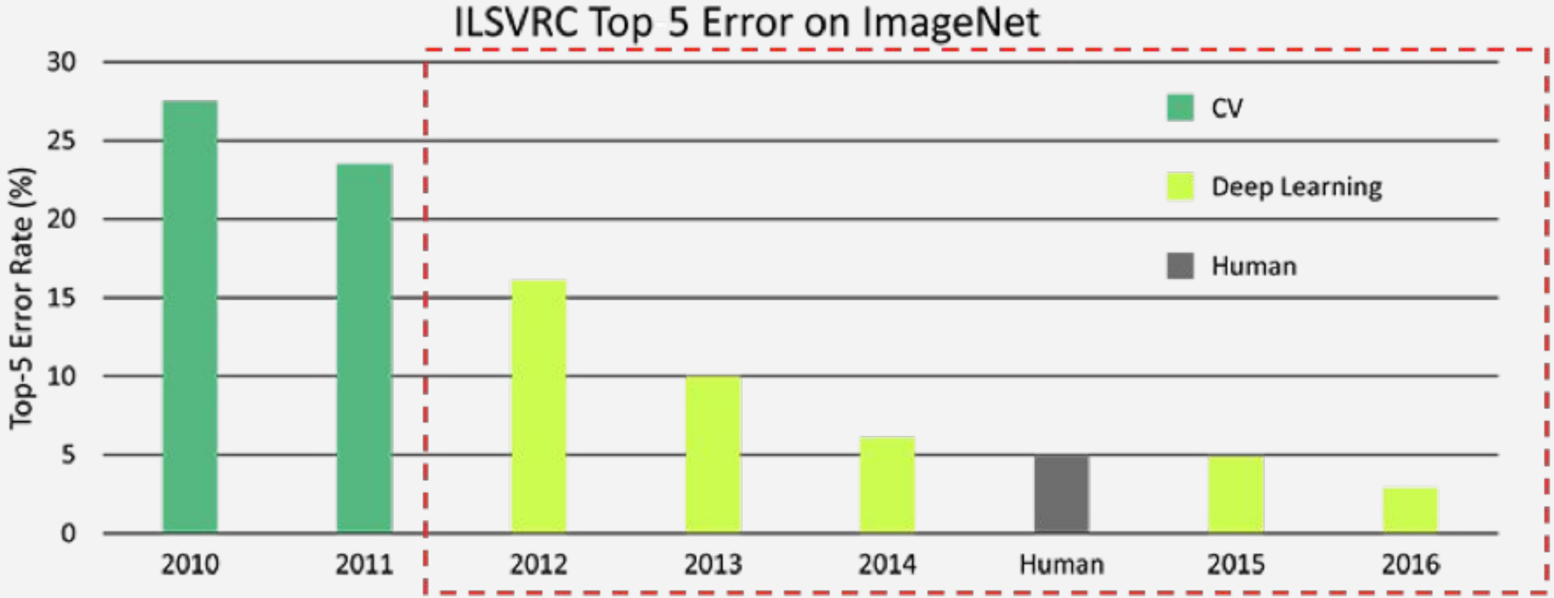


The Dominance of DL



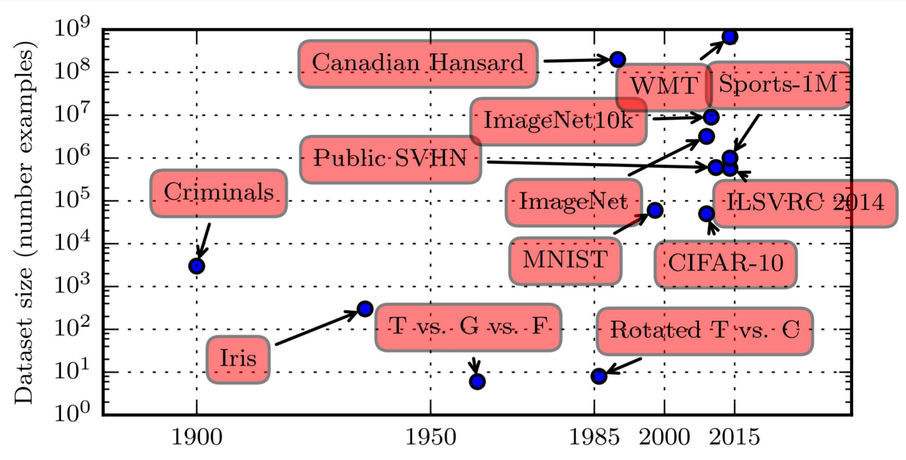
https://www.slideshare.net/roelofp/220115dlmeetup

The Dominance of DL



The introduction of Deep Learning techniques drove performance on image categorization from 30% error rates in 2010, down to <2% in 2017

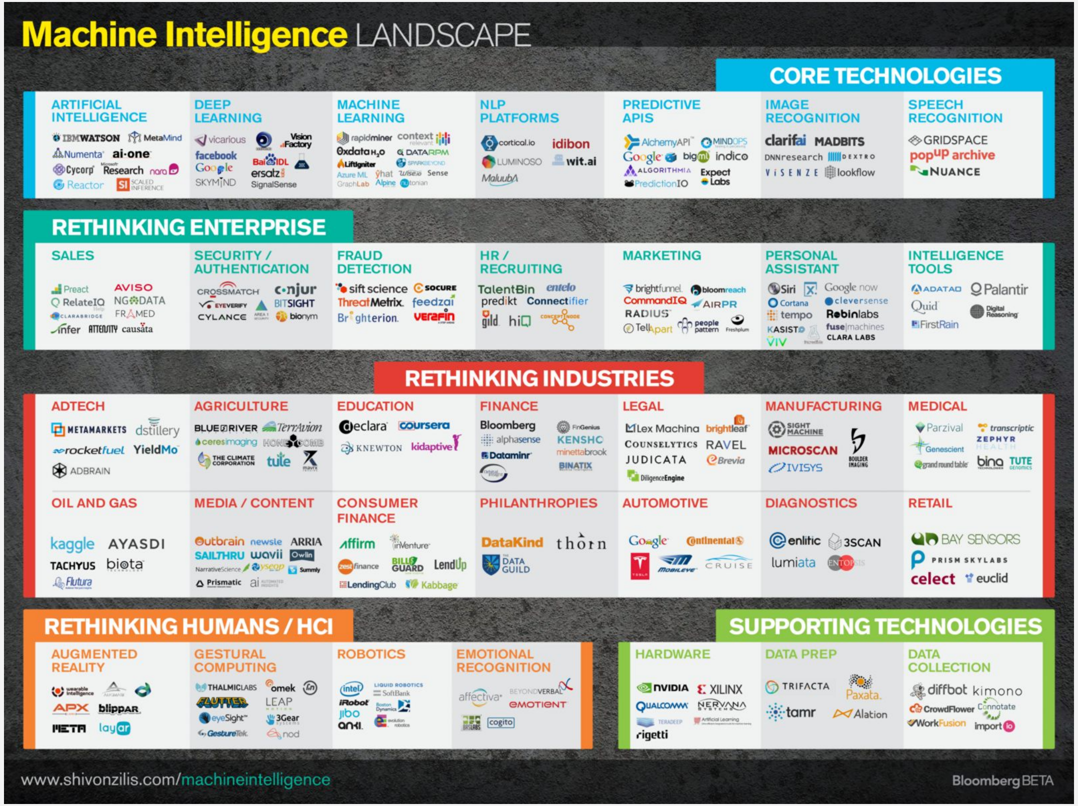
Main Reasons Behind Deep Learning's Success



Data

Hardware

Deep Learning Landscape



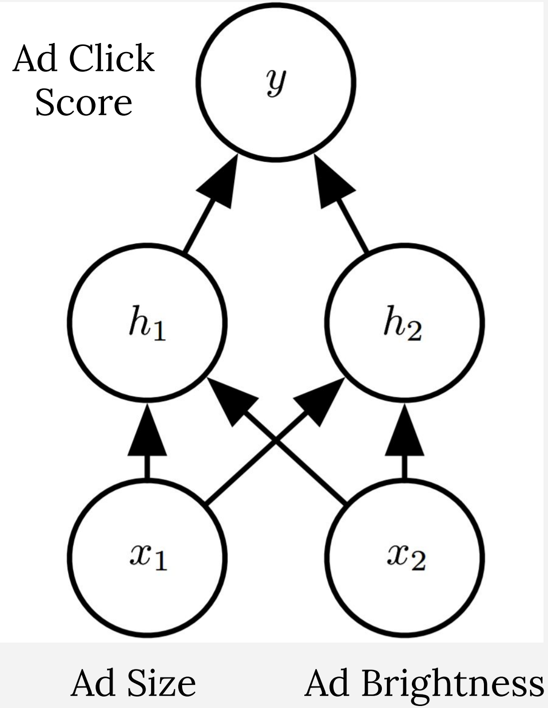
https://medium.com/@shivon/the-current-state-of-machine-intelligence-f76c20db2fe1

Today

- Overview
- **Basic Feedforward Networks and Core Concepts**
- Deeper Networks: Challenges and Solutions
- Convolutional Networks
- Recurrent Networks
- Generative Models
- Research Frontiers

Some slides adapted from Ian Goodfellow.

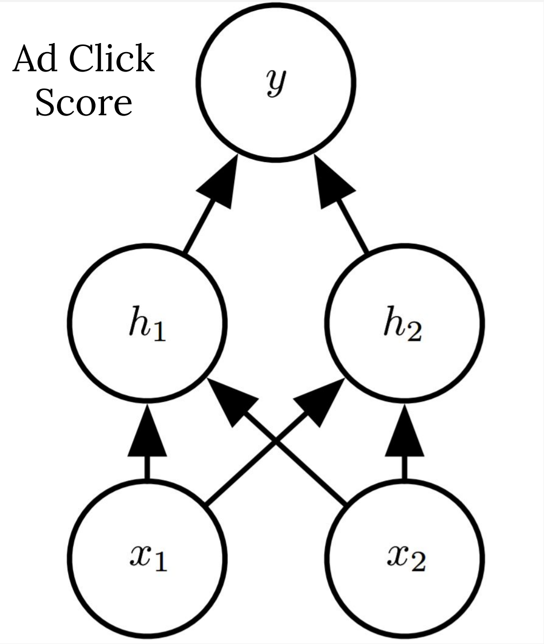
Basic Feedforward Networks



h_1 = x_1 * W_11 + x_2 * W_21 + c_1

h_2 = x_1 * W_12 + x_2 * W_22 + c_2

Basic Feedforward Networks



Ad Size Ad Brightness

Or any non-linear function

y = max(0, h_1) * w_1

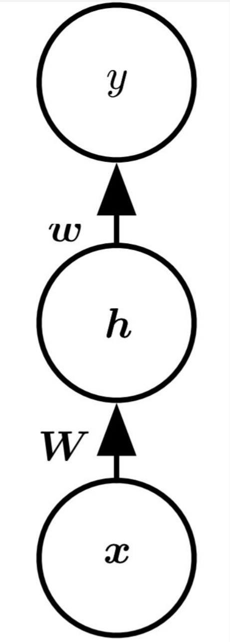
+ max(0, h_2) * w_2 + b

h_1 = x_1 * W_11 + x_2 * W_21 + c_1

h_2 = x_1 * W_12 + x_2 * W_22 + c_2

Basic **Feedforward** Networks

Ad Click
Score



Ad Size Ad Brightness

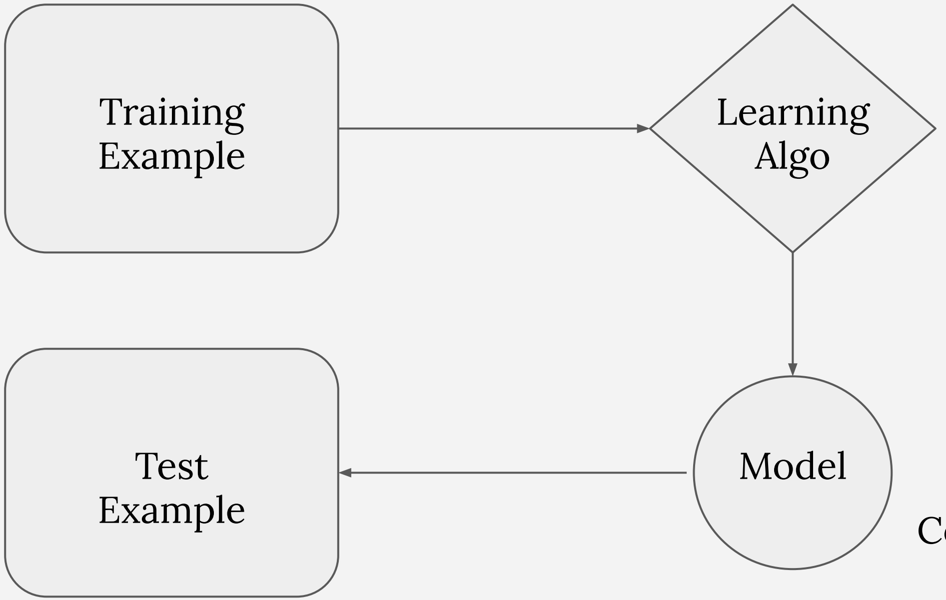
$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b.$$

Data

Network's learnable parameters

Recap: Machine Learning Paradigm

ham	Go until jurong point, crazy.. Availa
ham	Ok lar... Joking wif u oni...
spam	Free entry in 2 a wkly comp to win
ham	U dun say so early hor... U c alread
ham	Nah I don't think he goes to usf, he



Neural Network

ham	Even my brother is not like to spea
ham	As per your request 'Melle Melle ((
spam	WINNER!! As a valued network cus
spam	Had your mobile 11 months or mo

Contains learned values

Recap: Formal Definition of Learning

Data
(word freq)

$$e \doteq \mathop{\mathbb{E}}_{(x,y)\ \sim D}[L(y, f(x))]$$

Label
spam or not

Neural Network

$$\doteq \frac{1}{N}\sum_{n=1}^{N} L(y_n, f(x_n))$$

$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b.$$

Loss function
How good is our spam predictor?

Parameter Update: Gradient Descent

$$e \doteq \mathop{\mathbb{E}}_{(x,y)\ \sim D}[L(y, f(x))]$$

$$\doteq \frac{1}{N}\sum_{n=1}^{N} L(y_n, f(x_n))$$

$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b.$$

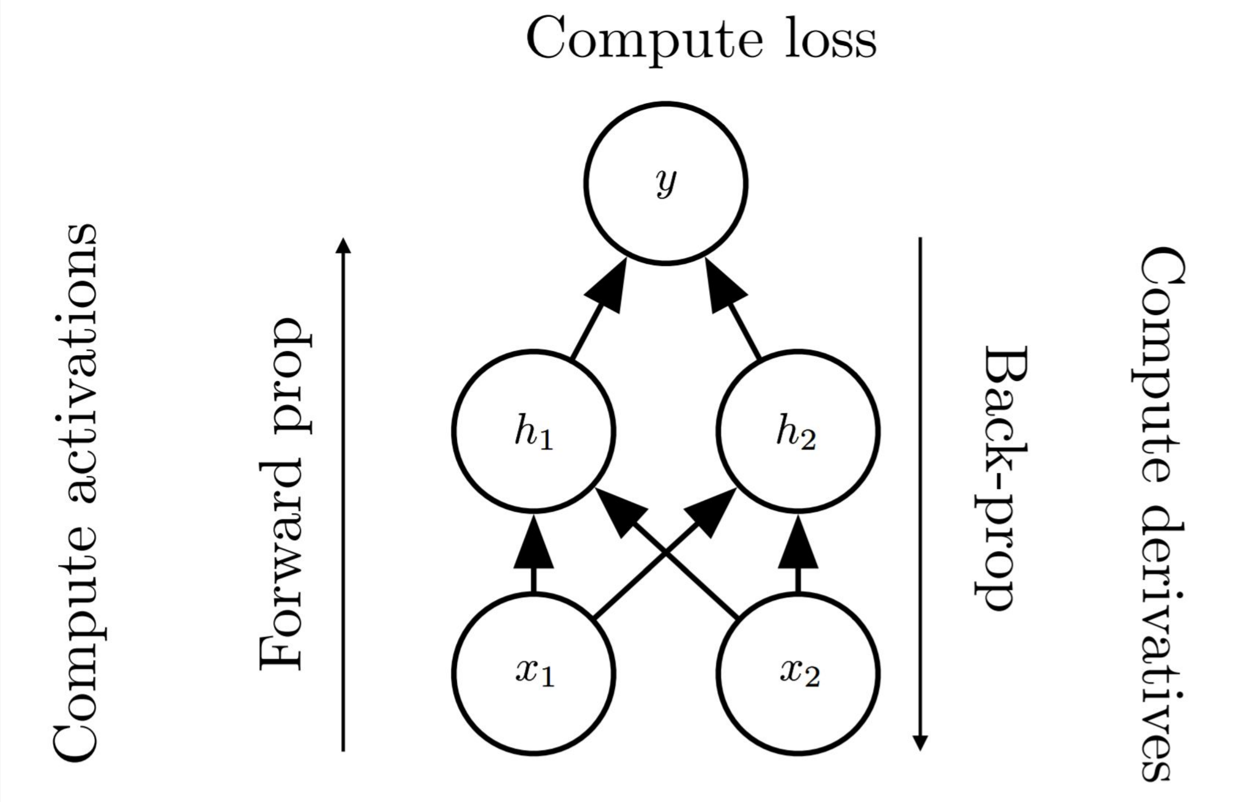
For t = 1 : T

$$W_{t+1} = W_t - \alpha \frac{\partial L}{\partial W_t}$$

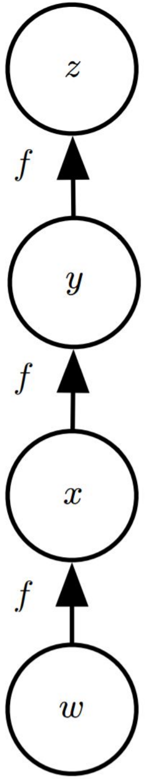
$$c_{t+1} = c_t - \alpha \frac{\partial L}{\partial c_t}$$

...

Backpropagation



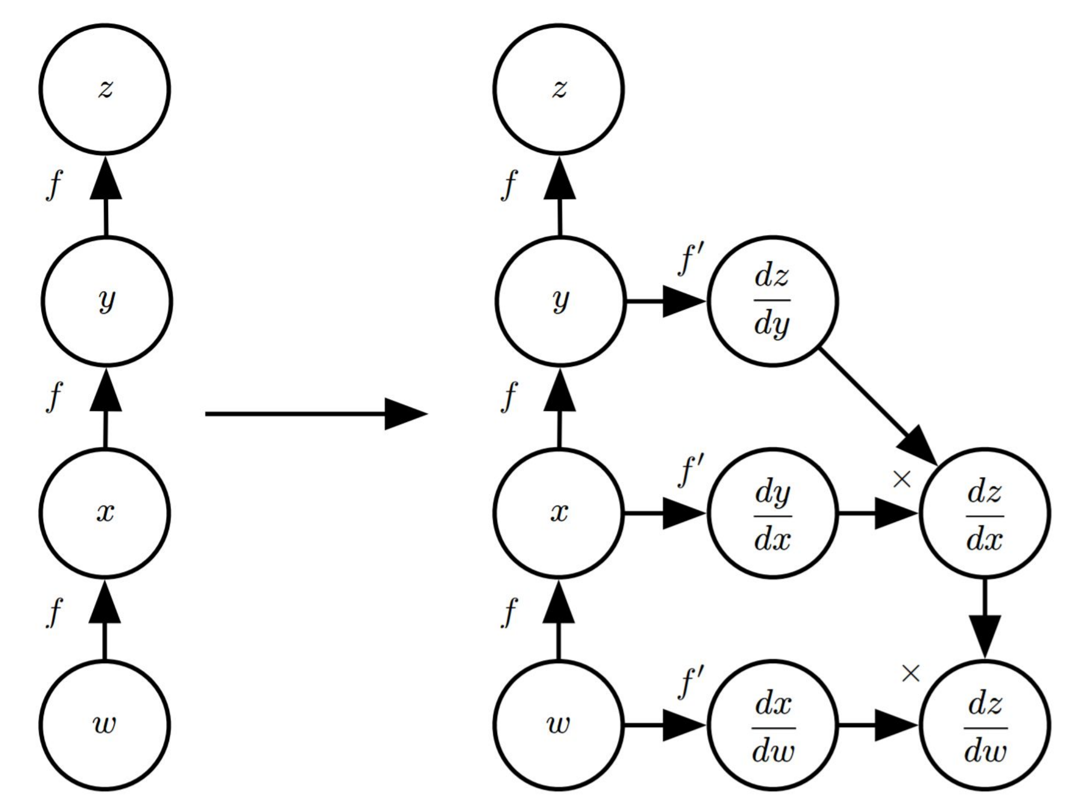
Backpropagation



$$
\begin{aligned}
&\frac{\partial z}{\partial w} \\
&= \frac{\partial z}{\partial y}\frac{\partial y}{\partial x}\frac{\partial x}{\partial w} \\
&= f'(y) f'(x) f'(w) \\
&= f'(f(f(w))) f'(f(w)) f'(w)
\end{aligned}
$$

Back-prop avoids computing this twice

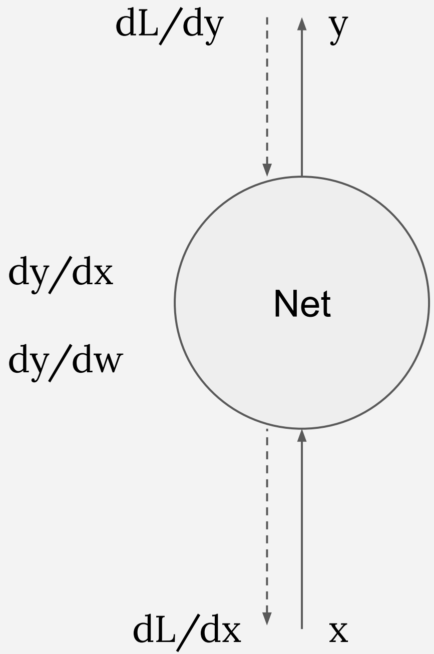
Backpropagation - How It Is Done



Backpropagation makes everything modular

Each network module only cares about its own information flow:

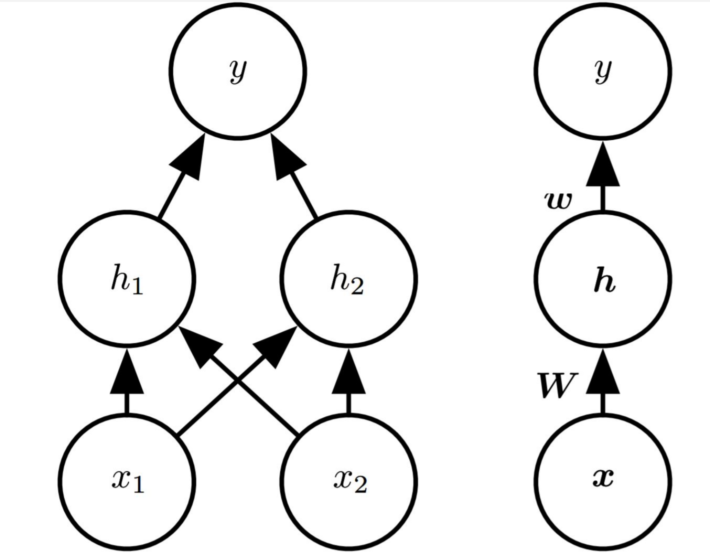
1. How to get from input to output
2. How to get loss from output to input
3. How to update weights



Backpropagation - PyTorch Example

```python
import torch
from torch.autograd import Variable

x = Variable(torch.rand(2,1), requires_grad=True)

W = Variable(torch.rand(2,2), requires_grad=True)
c = Variable(torch.rand(2,1), requires_grad=True)

w = Variable(torch.rand(2,1), requires_grad=True)
b = Variable(torch.rand(1), requires_grad=True)

h = torch.relu(torch.matmul(W, x) + c)
y = torch.matmul(w.t(),h) + b

y_target = Variable(torch.zeros(1))
loss  = (y - y_target)**2

loss.backward()
```

$$f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{c}, \boldsymbol{w}, b) = \boldsymbol{w}^\top \max\{0, \boldsymbol{W}^\top \boldsymbol{x} + \boldsymbol{c}\} + b.$$



Auto differentiation magic

Today

- Overview
- Basic Feedforward Networks and Core Concepts
- **Deeper Networks: Challenges and Solutions**
- Convolutional Networks
- Recurrent Networks
- Generative Models
- Research Frontiers

Some slides adapted from [Ian Goodfellow].

Training with deep neural networks

Backpropagation should work with any number of layers.

Can we just stack arbitrary number of layers and form a really deep thus powerful neural network?

It turned out there are many challenges.

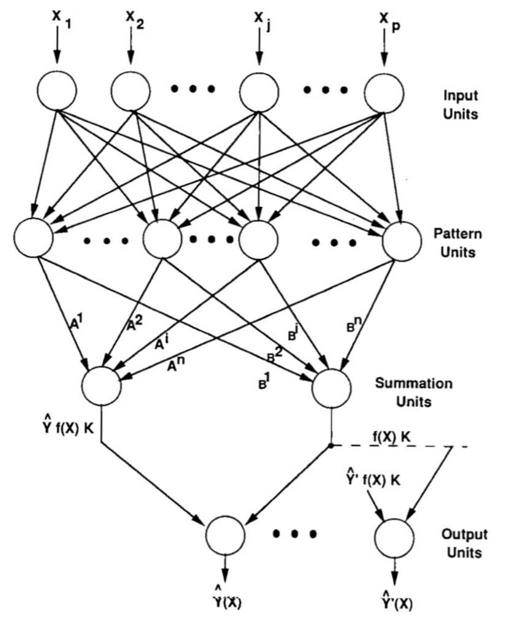


Fig. 1. GRNN block diagram.

Challenges with deep neural networks

To make efficient computation,

neural networks are running on GPUs.

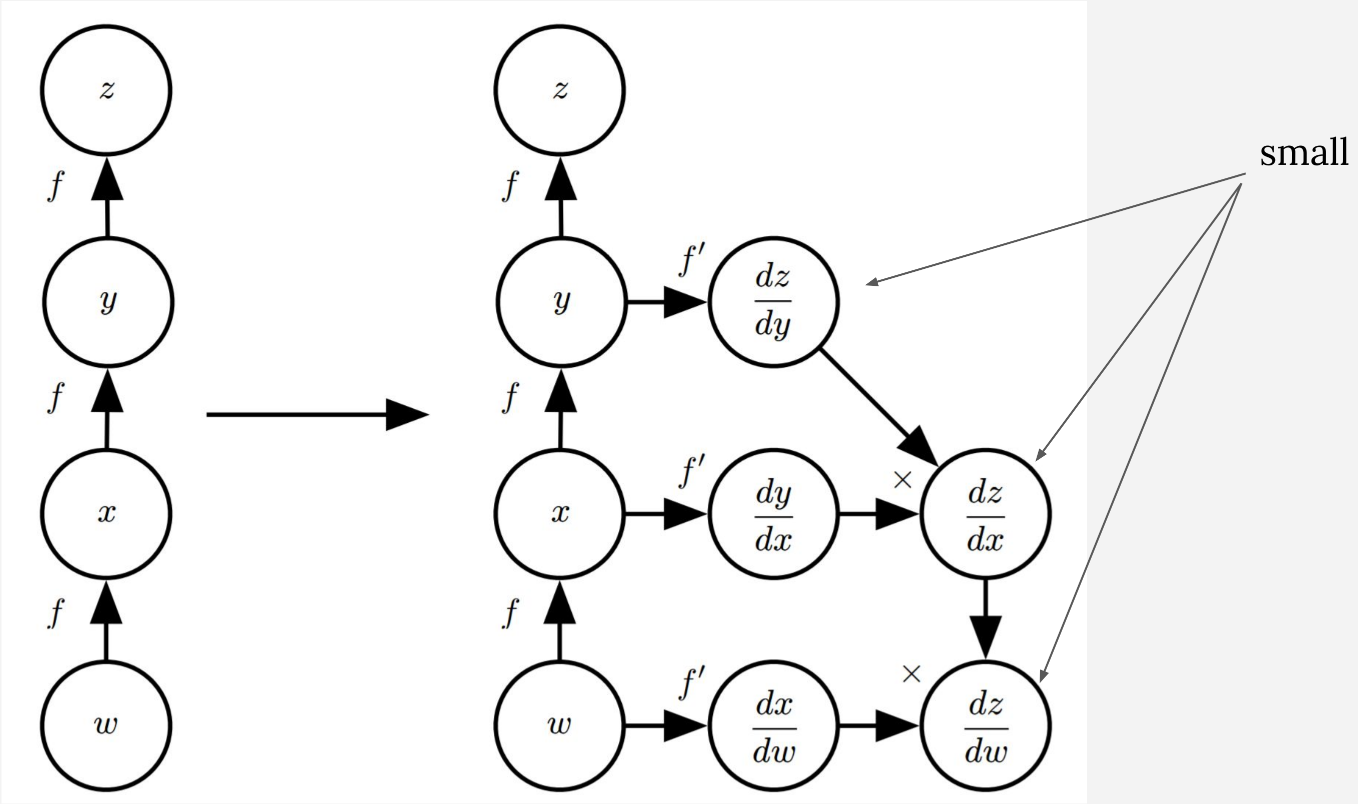
There is a hard limit on the GPU memory

you can fit.

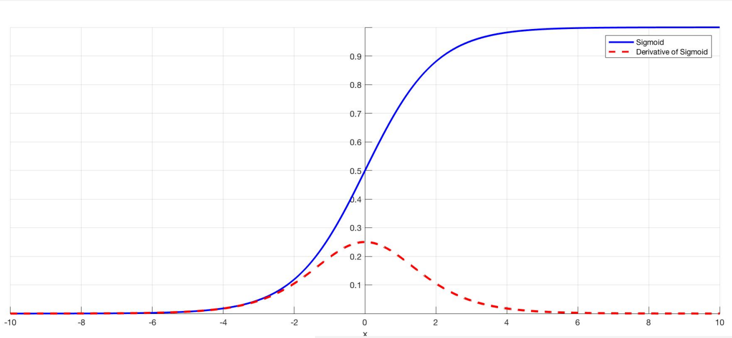
Hardware

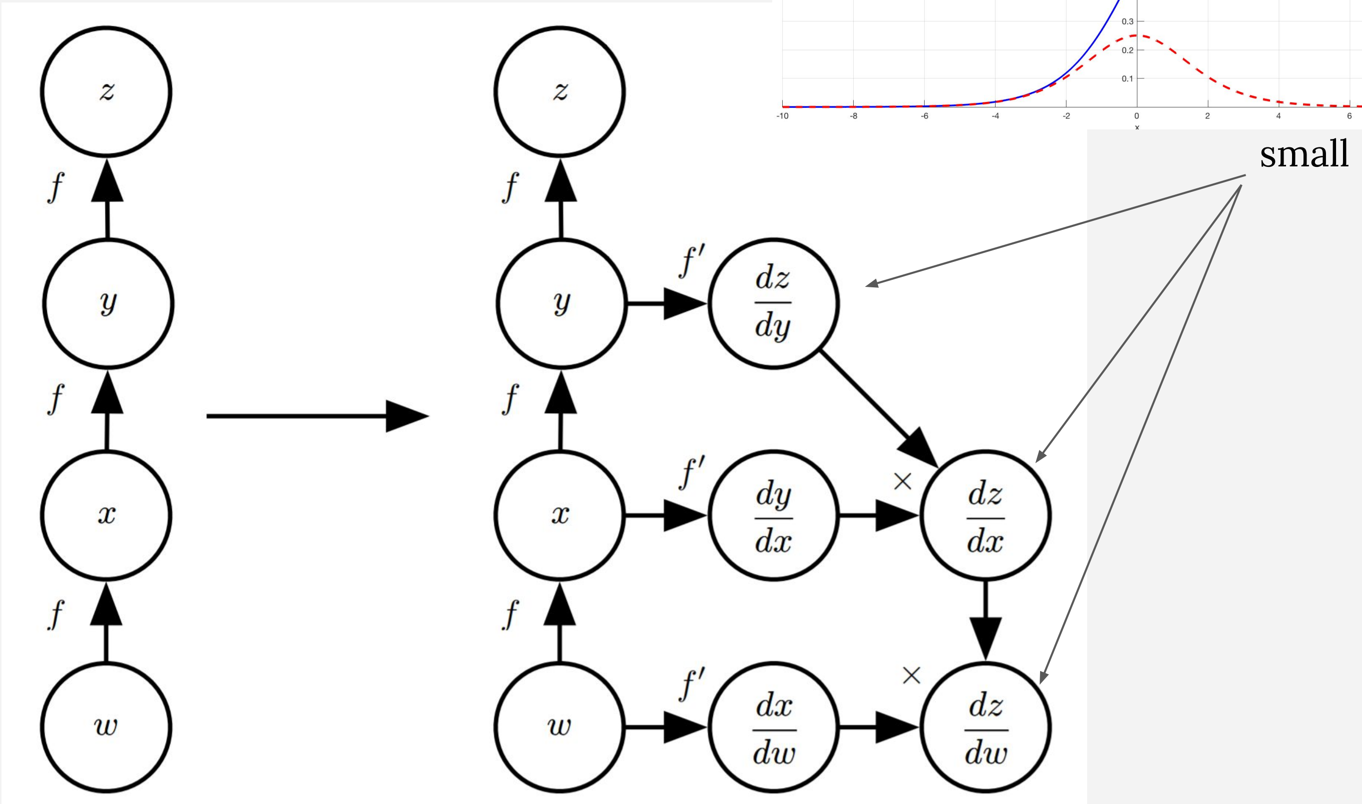
Distributed computing with many GPUs is possible. But it requires special communication/scheduling and learning algorithms.

Vanishing Gradient Problem

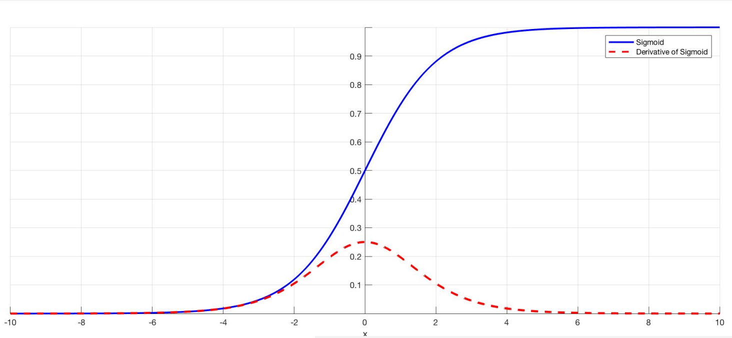


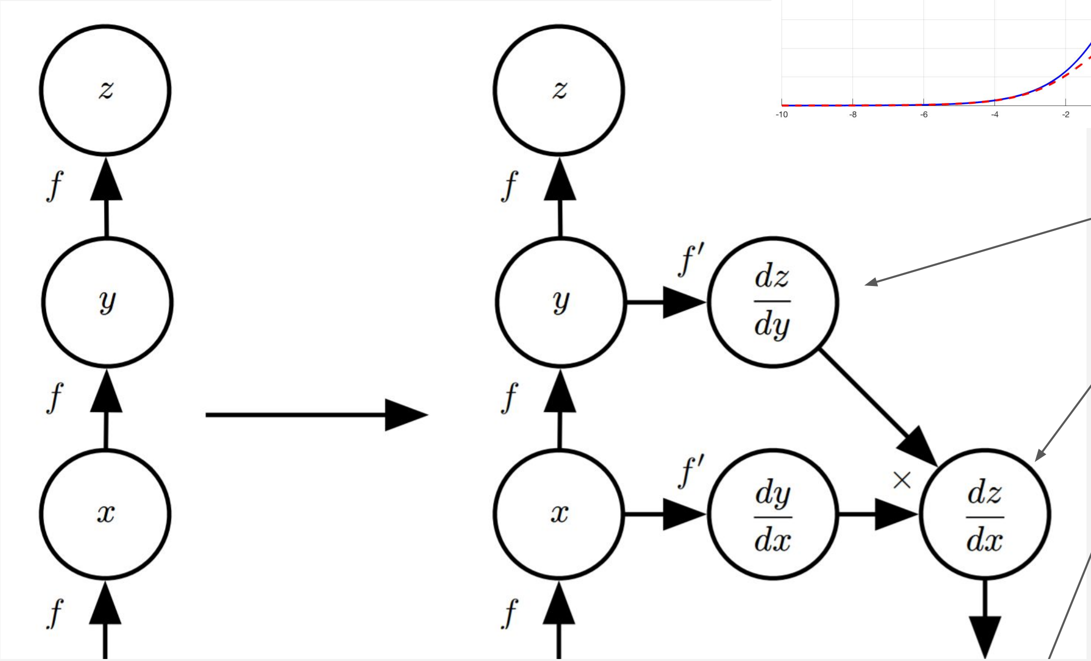
Vanishing Gradient Problem





Vanishing Gradient Problem



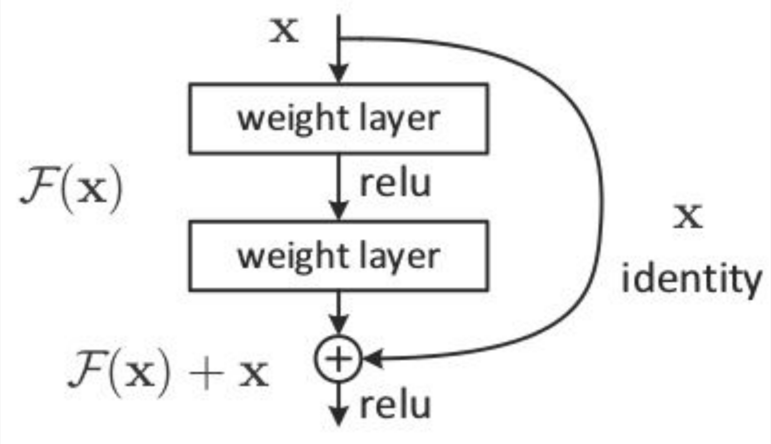


small

Gradient becomes too small for earlier layers -> no learning for them

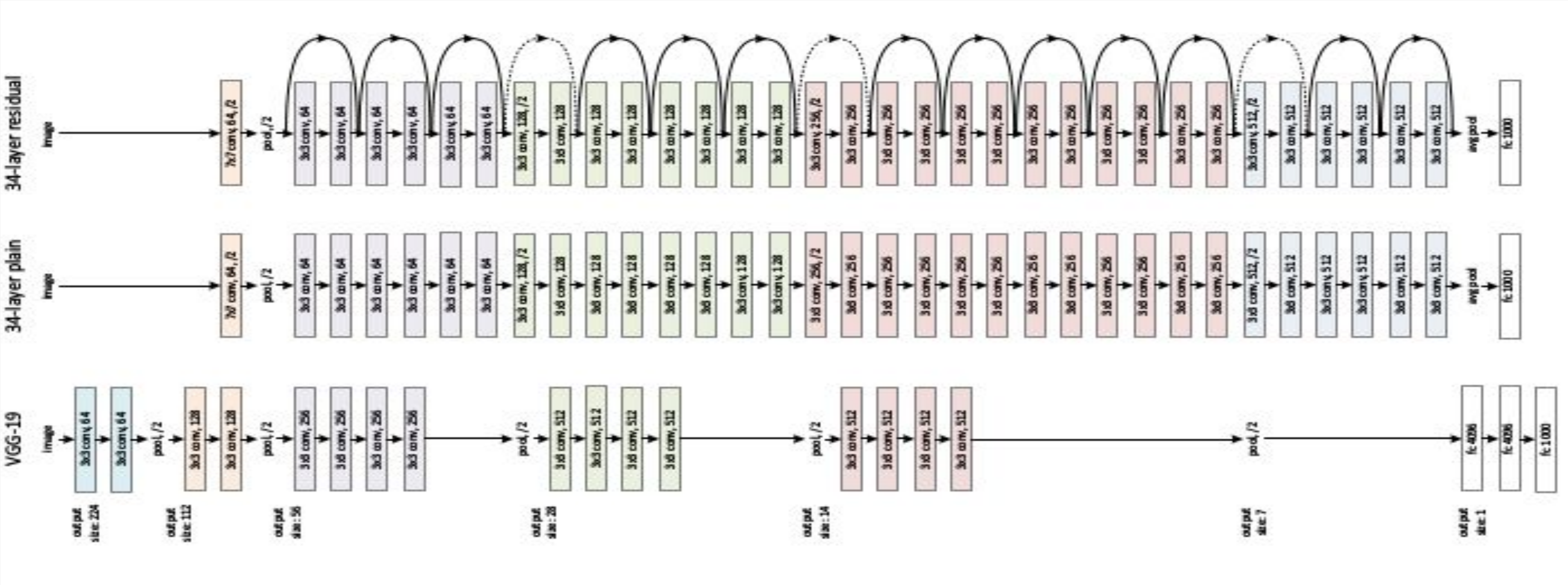
Solutions: ResNet

Residual Network Basic Block



Always has gradients larger than 0

Solutions: ResNet



Solutions: BatchNorm

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

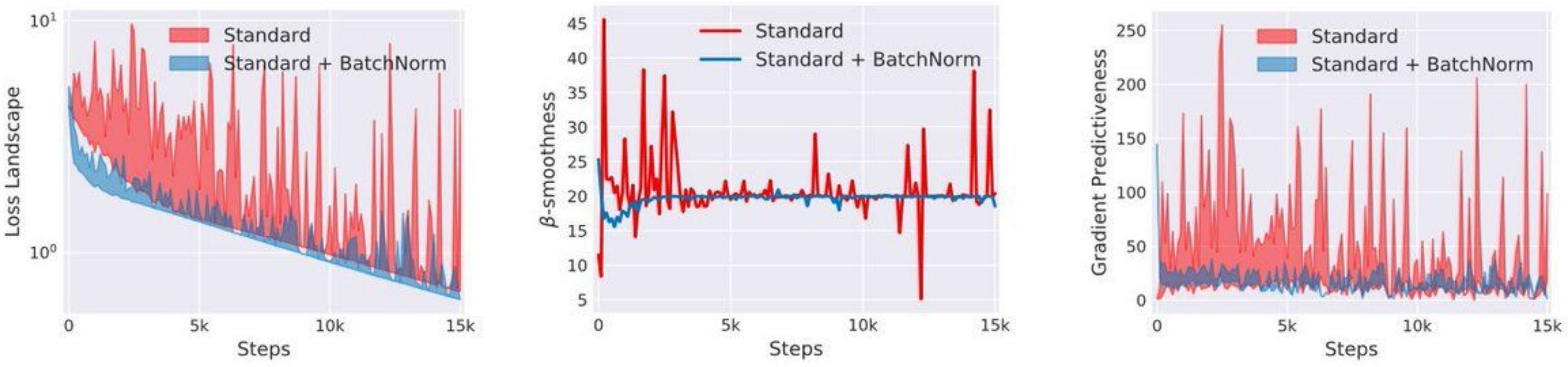
$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

https://arxiv.org/pdf/1502.03167.pdf

Solutions: BatchNorm

Stabilize training, prevent gradient exploding / vanishing.



(a) loss landscape (b) "effective" β-smoothness (c) gradient predictiveness

Santurkar, Shibani, et al. "How does batch normalization help optimization?." *Advances in Neural Information Processing Systems*. 2018.

Deep Neural Networks are prone to overfitting

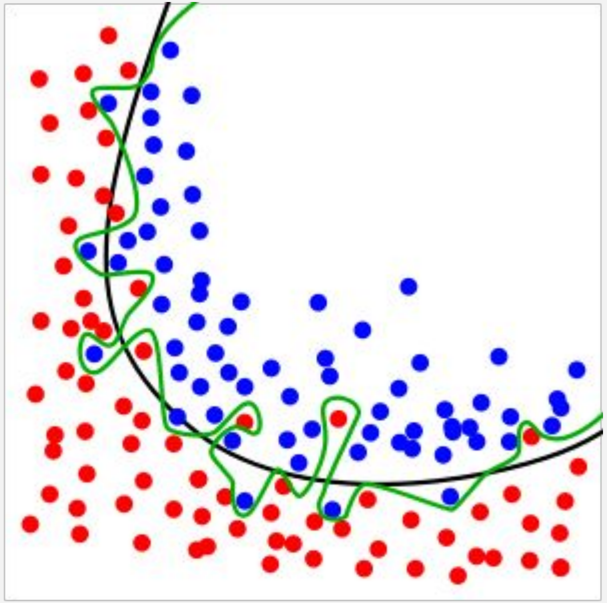
Regularization

L1 Norm

Prefer sparse weights

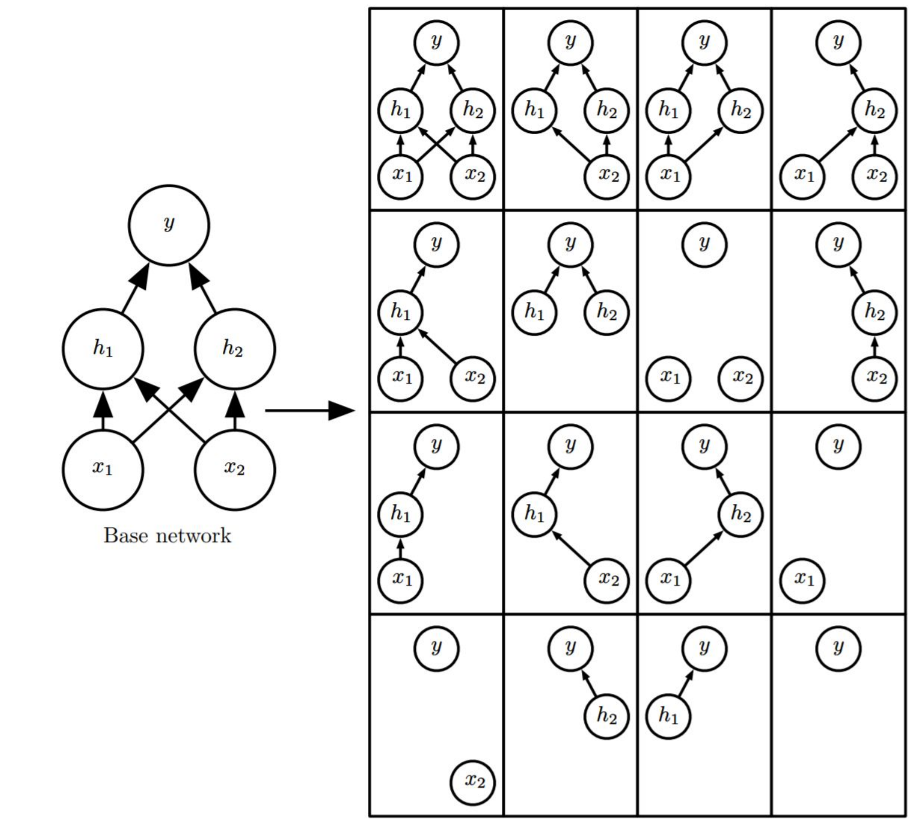
L2 Norm

Prefer smaller weights



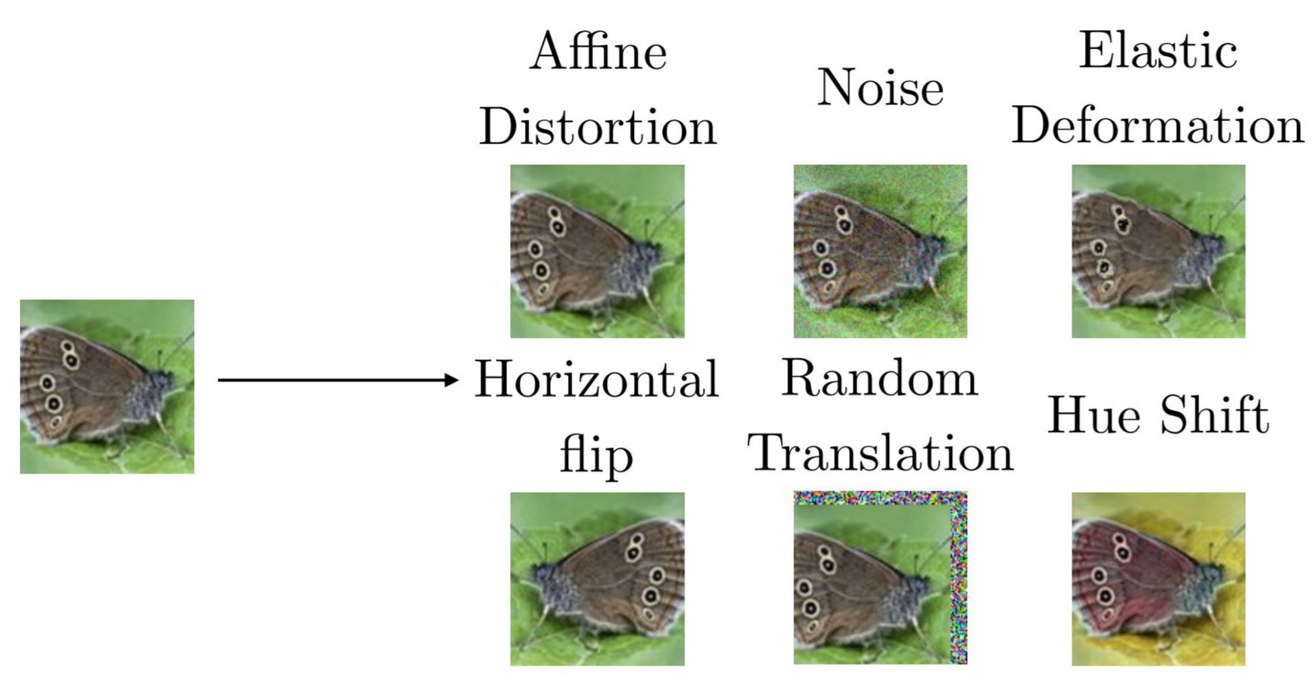
Regularization

Dropout



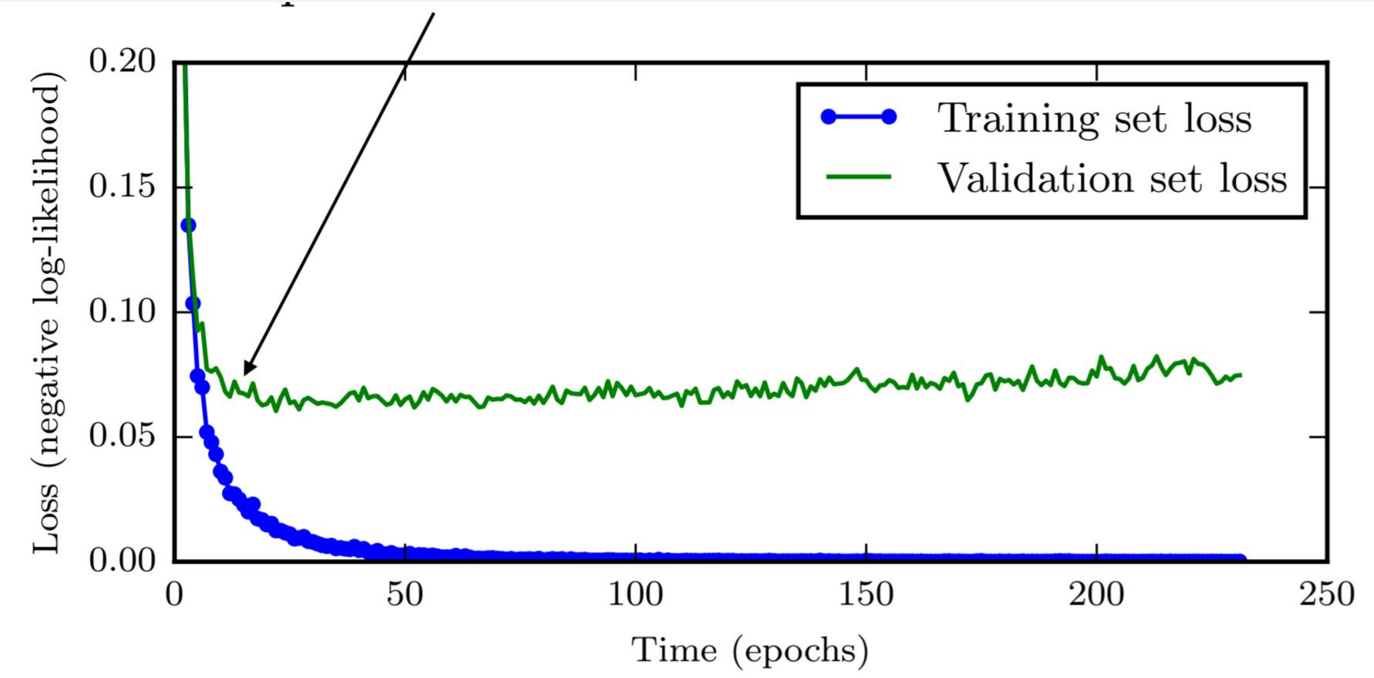
Regularization

Data Augmentation



Regularization

Early Stopping



Today

- Overview
- Basic Feedforward Networks and Core Concepts
- Deeper Networks: Challenges and Solutions
- **Convolutional Networks**
- Recurrent Networks
- Generative Models
- Research Frontiers

Data and Neural Network Models

Static Data

Convolutional Neural Networks

Dynamic Data

Recurrent Neural Networks

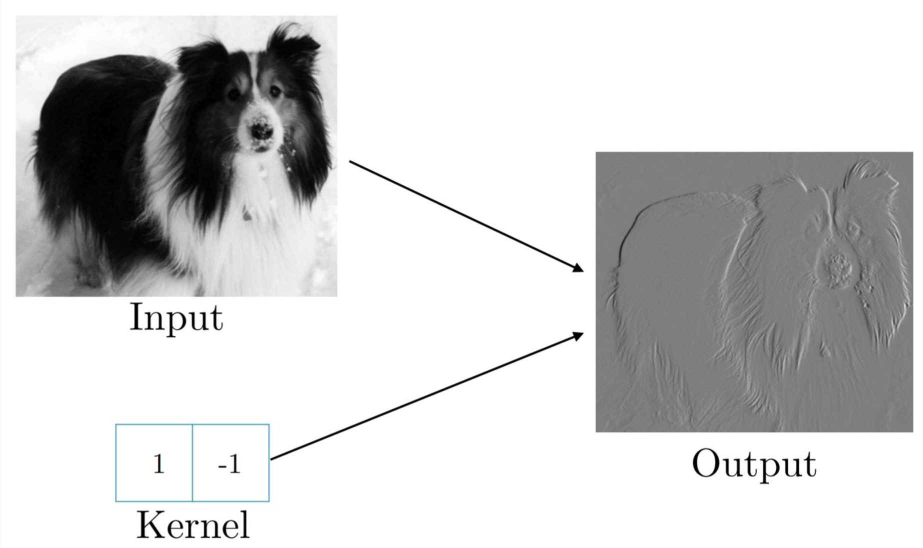
Unsupervised Data

Generative Neural Networks

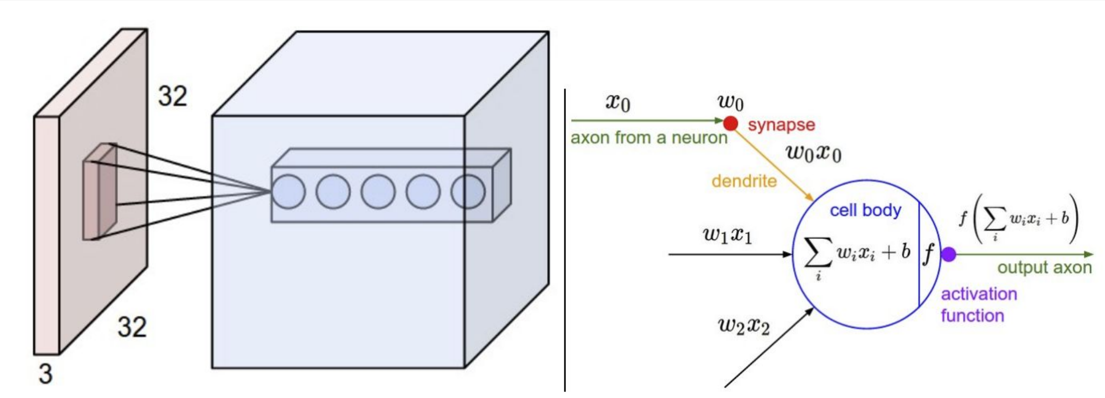
Convolutional Networks

Convolution

A local operation that extracts information from data.



Convolutional Networks

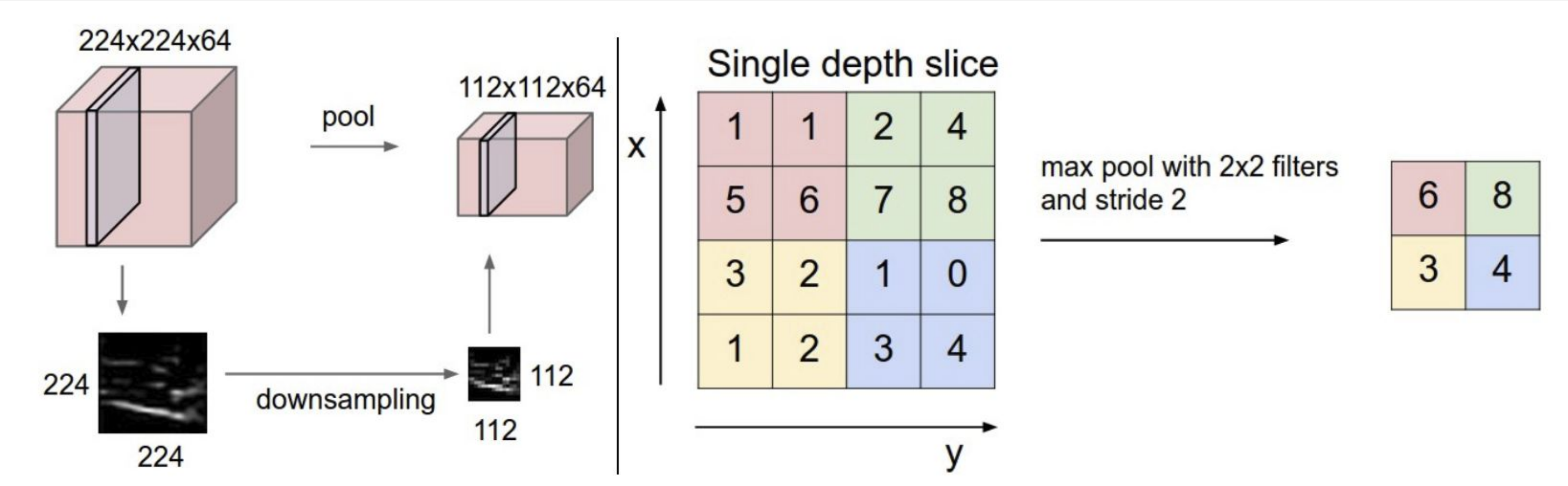


Convolution in action:

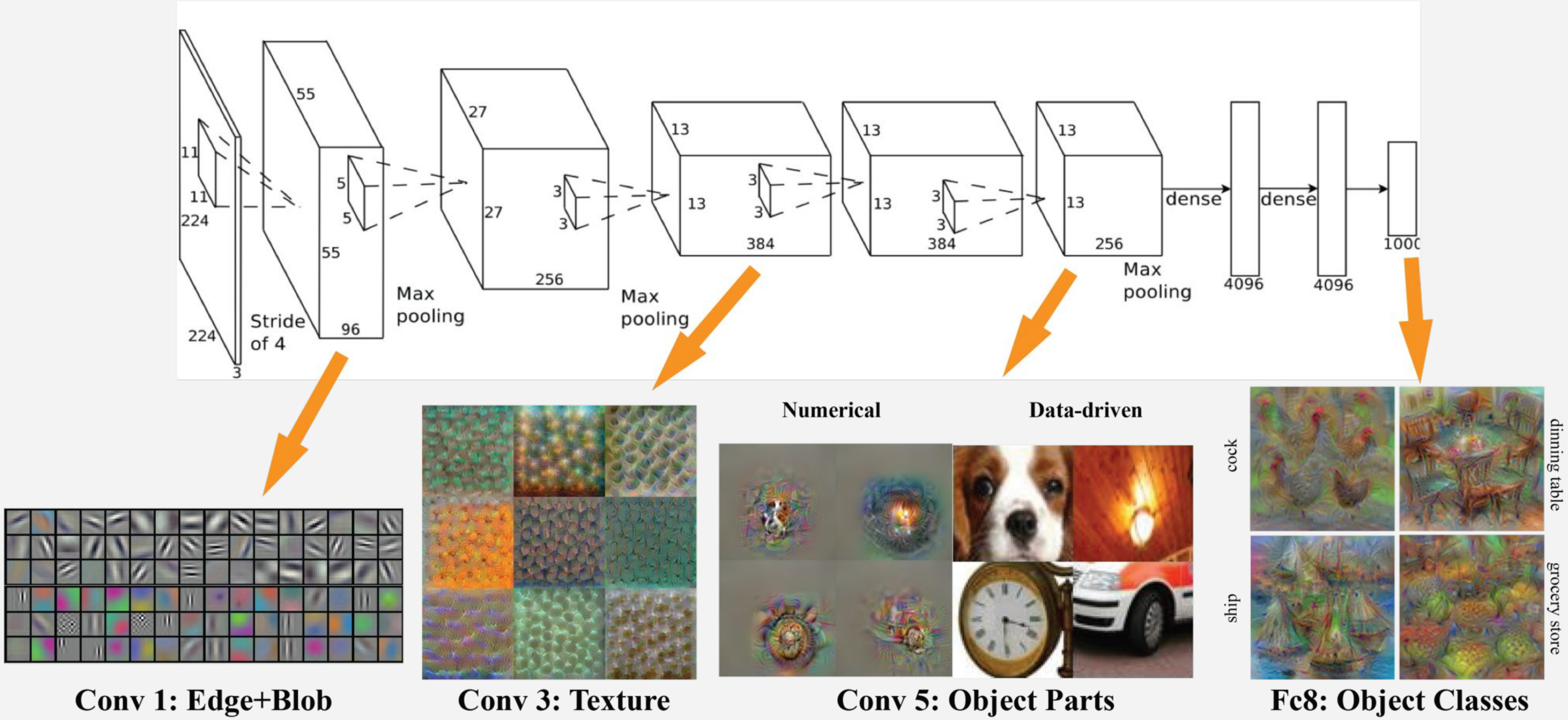
http://cs231n.github.io/convolutional-networks/

Convolutional Networks

Pooling



Convolutional Networks



Convolutional Networks

Good for:

Data with translation invariance and shared statistics.

Data that can benefitted from different levels of abstraction.

Not so good for:

Dynamic data.

Today

- Overview
- Basic Feedforward Networks and Core Concepts
- Deeper Networks: Challenges and Solutions
- Convolutional Networks
- **Recurrent Networks**
- Generative Models
- Research Frontiers

Data and Neural Network Models

Static Data	Dynamic Data	Unsupervised Data
Convolutional Neural Networks	Recurrent Neural Networks	Generative Neural Networks

Recurrent Networks

Dynamic Data

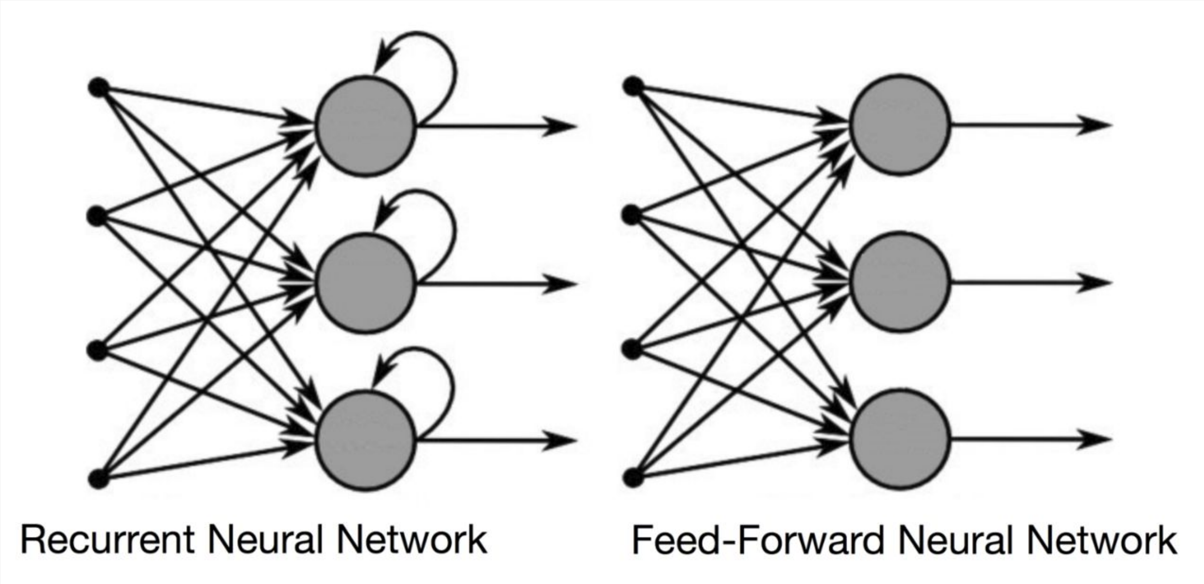
Data changes over time.

Language

Video

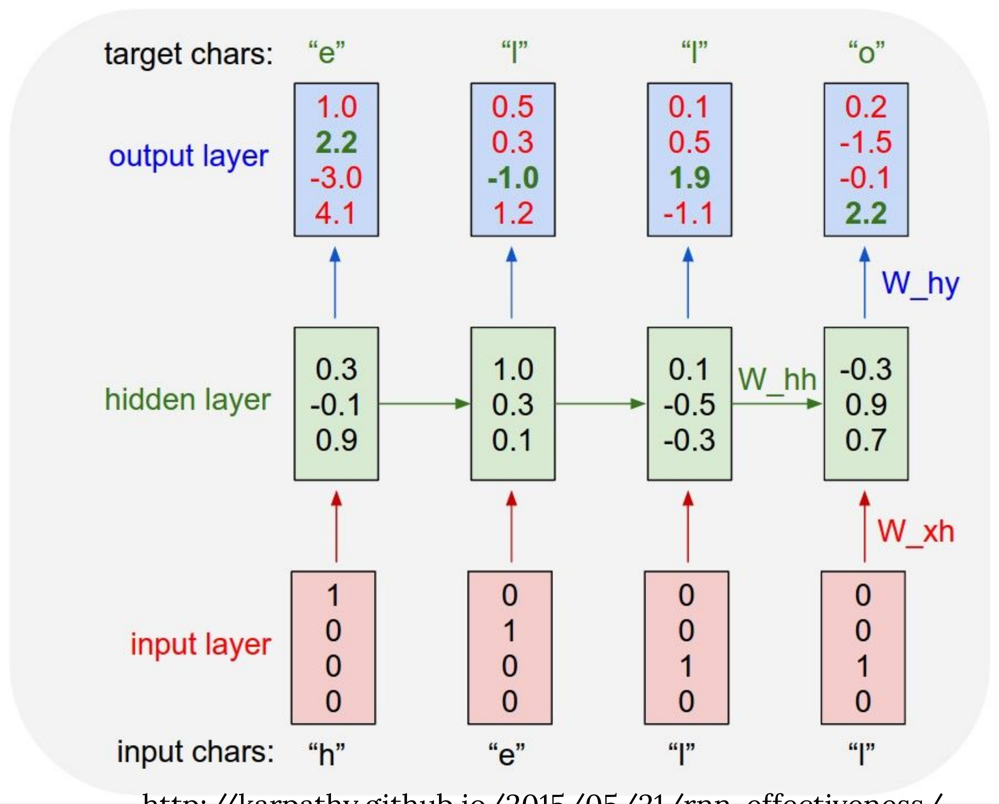
Recurrent Networks

A model's output is not just depending on the current input



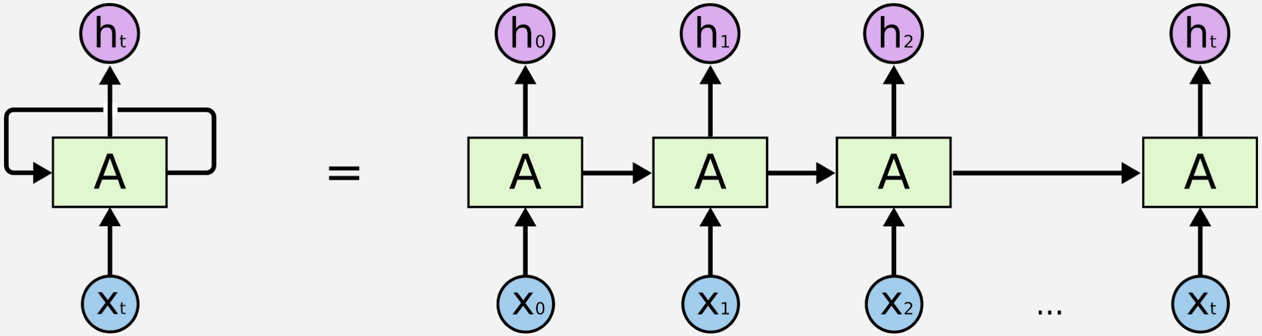
https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5

Recurrent Networks

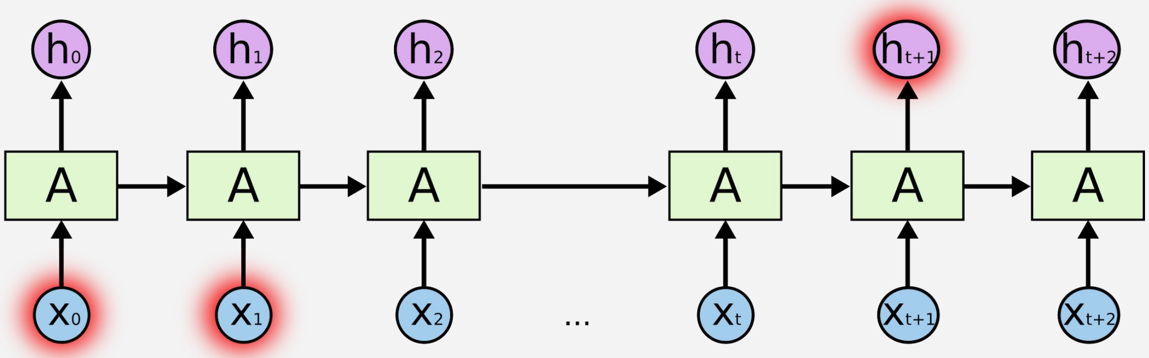


http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Unrolled RNNs



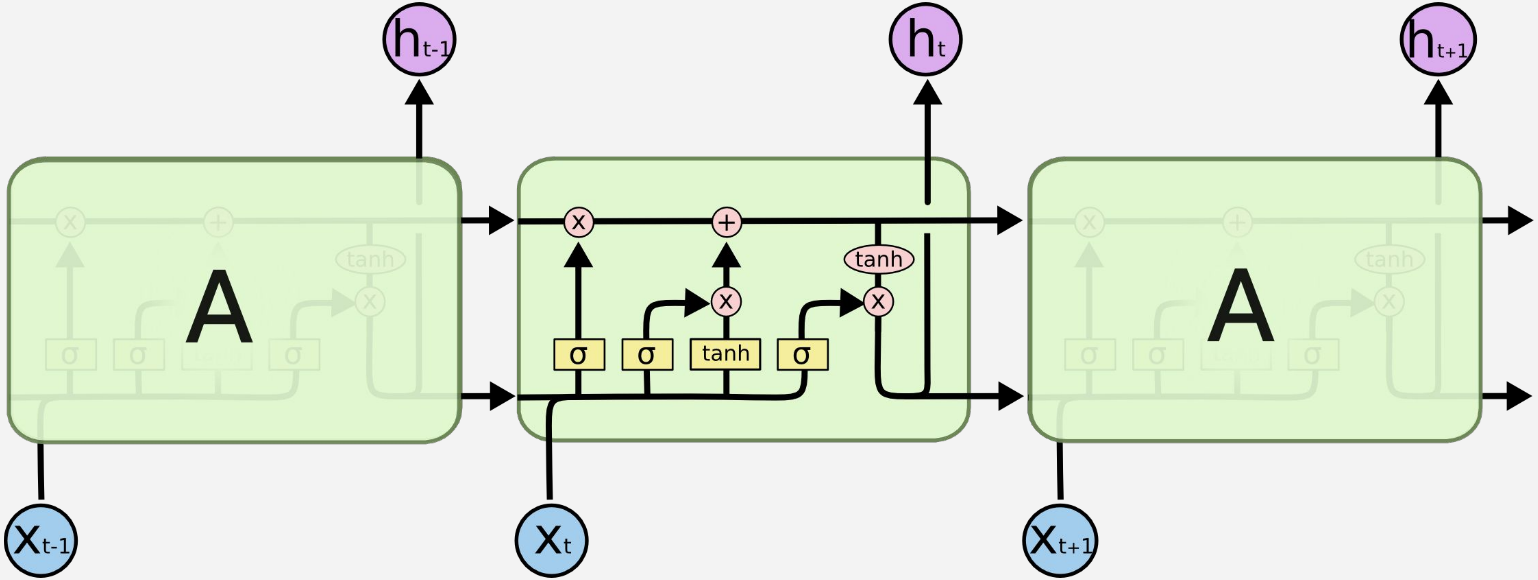
Long term relations



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short-Term Memory (LSTM) Networks

Being able to remember... and forget!



Recurrent Networks

Good for:

Dynamic data.

Not so good:

Might be tricky to train.

An interesting read.

Today

- Overview
- Basic Feedforward Networks and Core Concepts
- Deeper Networks: Challenges and Solutions
- Convolutional Networks
- Recurrent Networks
- **Generative Models**
- Research Frontiers

Data and Neural Network Models

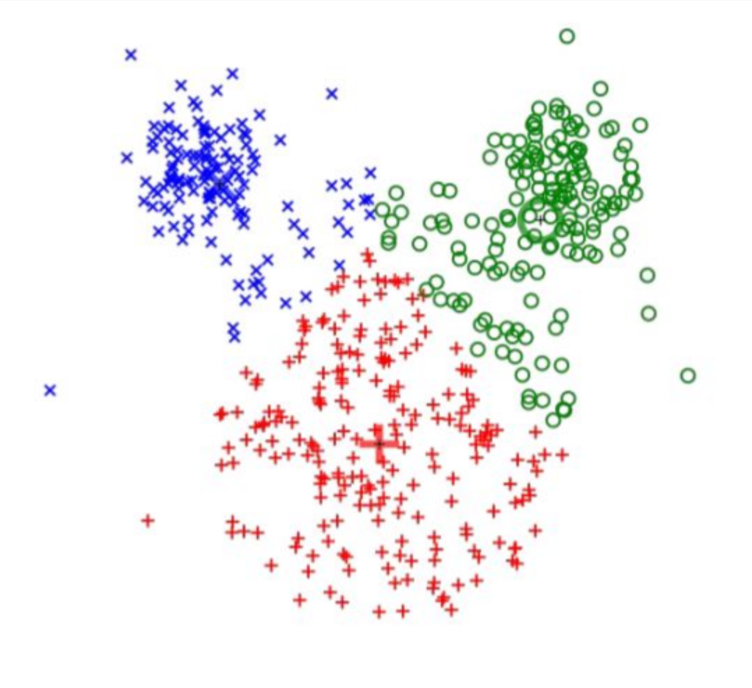
Static Data	Dynamic Data	Unsupervised Data
Convolutional Neural Networks	Recurrent Neural Networks	Generative Neural Networks

Generative Models

We have data, but no labels.

Goal

Recover underlying structures of the data.

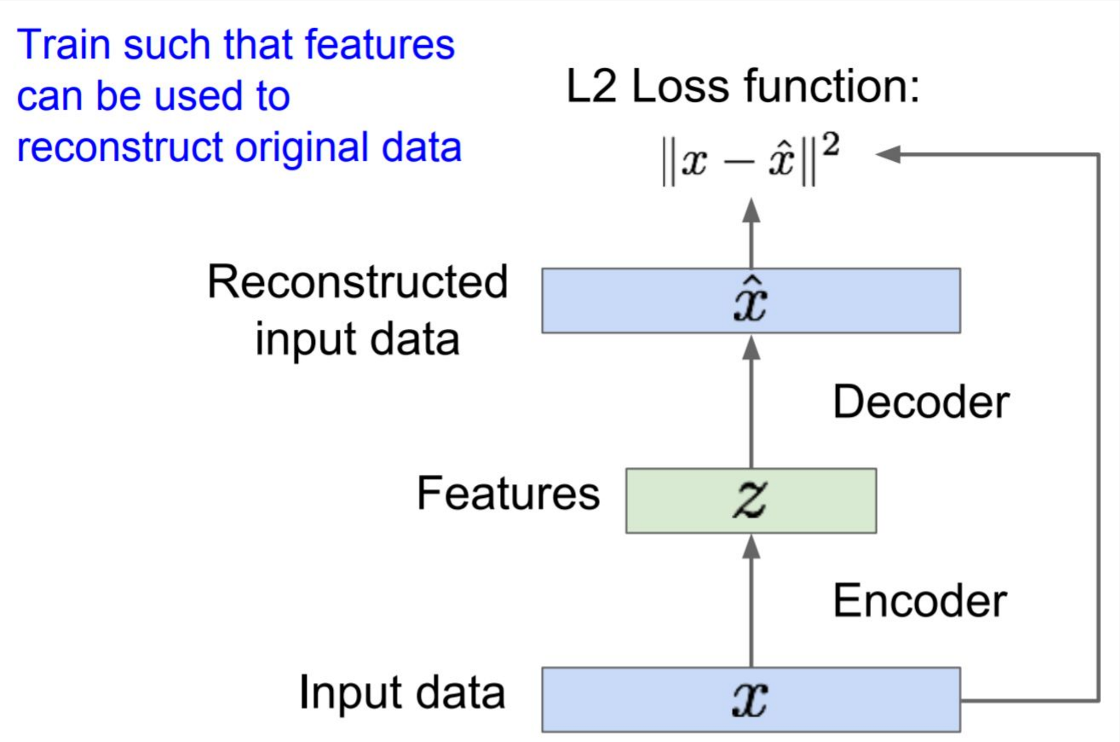


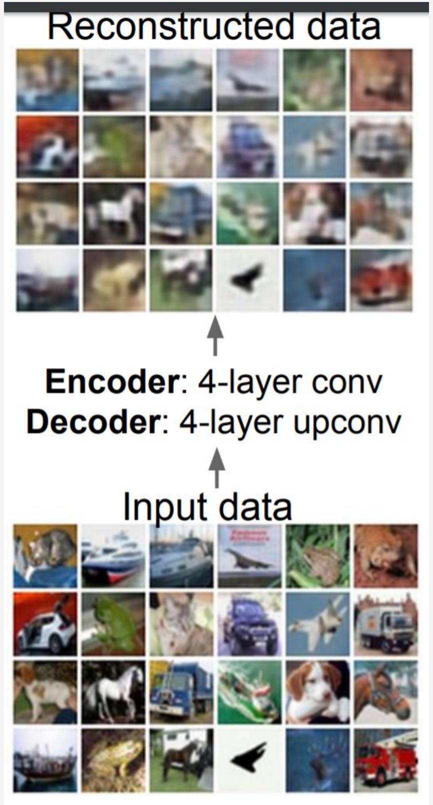
Generative Models

- Realistic samples for artwork, super-resolution, colorization, etc.

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Autoencoder

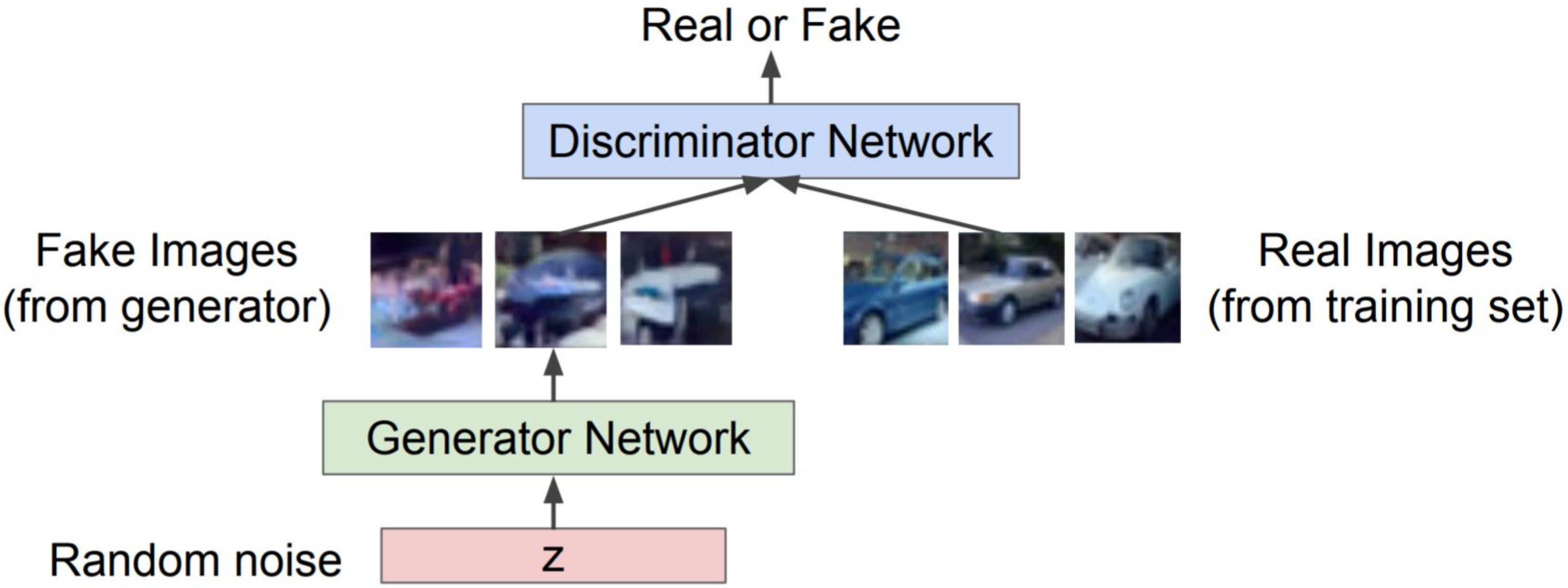




http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Generative Adversarial Networks

Generator network: try to fool the discriminator by generating real-looking images
Discriminator network: try to distinguish between real and fake images



http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture13.pdf

Today

- Overview
- Basic Feedforward Networks and Core Concepts
- Deeper Networks: Challenges and Solutions
- Convolutional Networks
- Recurrent Networks
- Generative Models
- **Research Frontiers**

Research Frontiers

Deeper and Bigger Networks

Distributed training

More Efficient Networks

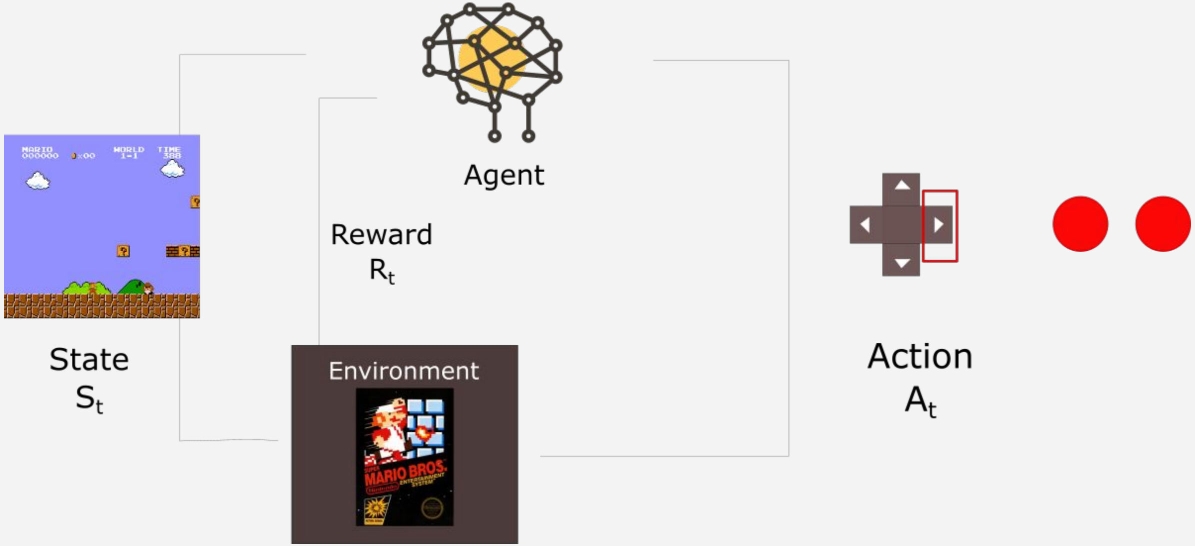
Network compression / Binary networks

Understanding Networks

The explainability of neural networks

Research Frontiers

Reinforcement Learning is trying to solve a very different problem than standard ML: instead of supervision, we are given a vague signal called 'reward'.



https://medium.freecodecamp.org/an-introduction-to-reinforcement-learning-4339519de419

Summary

- Overview
- Basic Feedforward Networks and Core Concepts
- Deeper Networks: Challenges and Solutions
- Convolutional Networks
- Recurrent Networks
- Generative Models
- Research Frontiers

Further Readings:

Deep Learning by Ian Goodfellow and Yoshua Bengio and Aaron Courville link