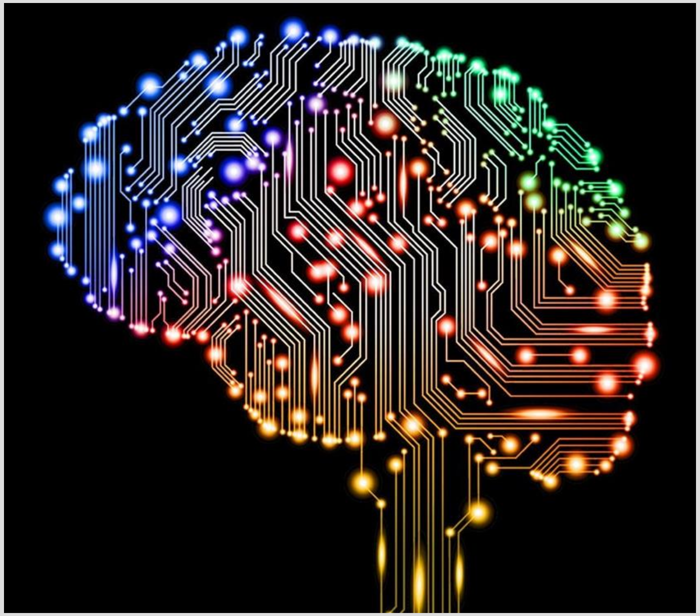
Neural Networks - Deep Learning and Transfer Learning

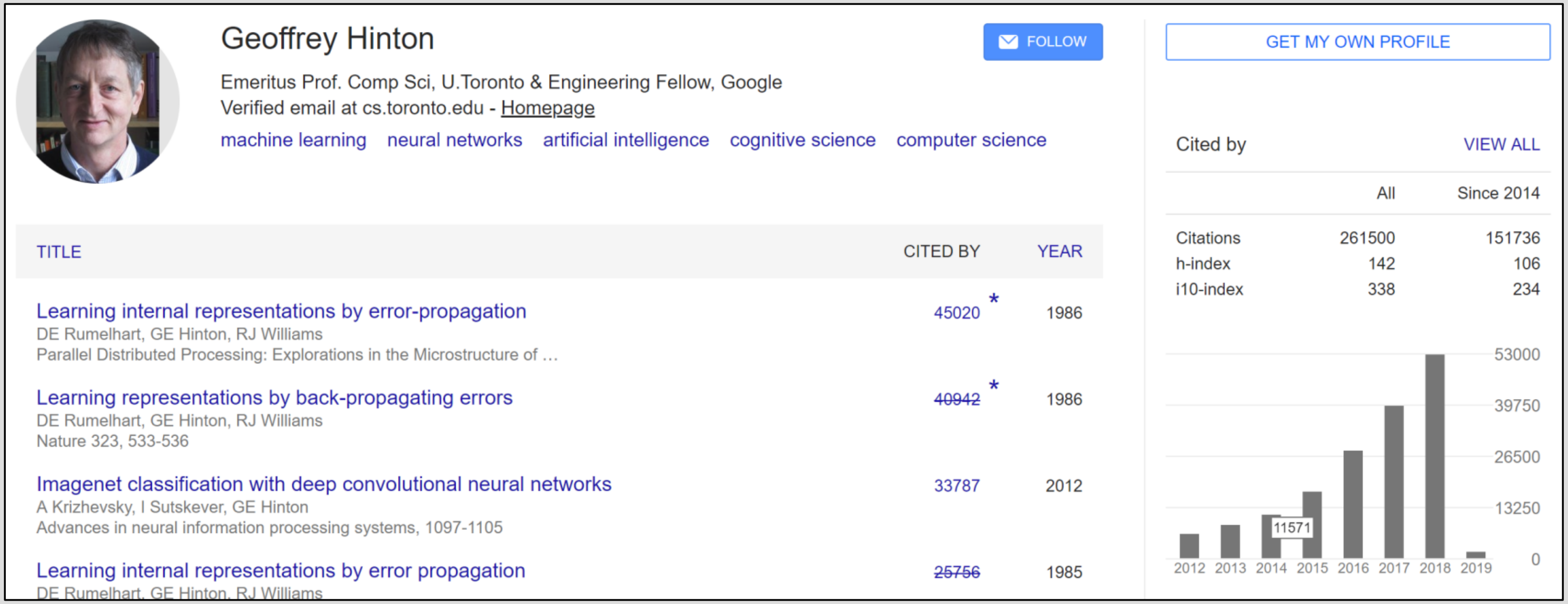
Topics

- Deep learning
- Convolutional neural networks
- Transfer learning
- Augmentation

Deep Learning

- Biggest thing in “AI” for awhile
- Inspired in part by “deep neural architectures” (human brain)
- Citation crazy





Geoffrey Hinton

FOLLOW

Emeritus Prof. Comp Sci, U.Toronto & Engineering Fellow, Google
Verified email at cs.toronto.edu - Homepage
machine learning neural networks artificial intelligence cognitive science computer science

TITLE	CITED BY	YEAR
Learning internal representations by error-propagation DE Rumelhart, GE Hinton, RJ Williams Parallel Distributed Processing: Explorations in the Microstructure of …	45020 *	1986
Learning representations by back-propagating errors DE Rumelhart, GE Hinton, RJ Williams Nature 323, 533-536	~~40942~~ *	1986
Imagenet classification with deep convolutional neural networks A Krizhevsky, I Sutskever, GE Hinton Advances in neural information processing systems, 1097-1105	33787	2012
Learning internal representations by error propagation DE Rumelhart, GE Hinton, RJ Williams	~~25756~~	1985

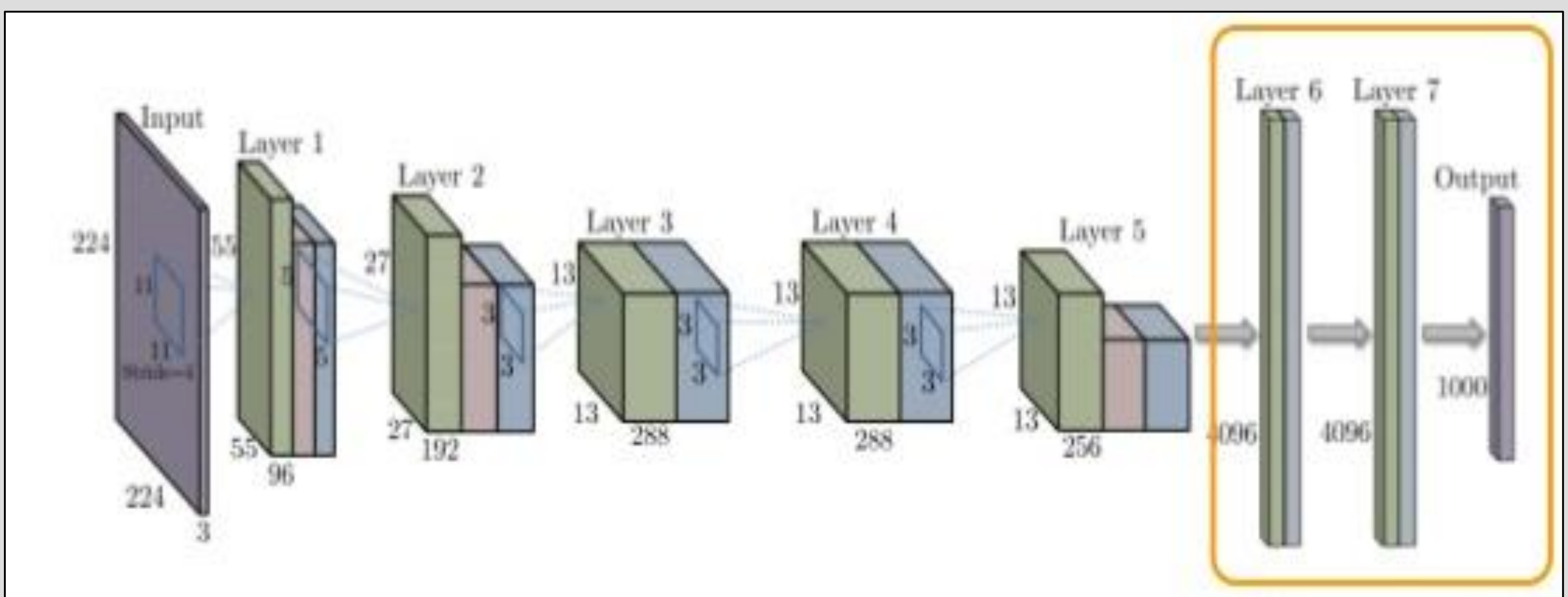
GET MY OWN PROFILE

Cited by VIEW ALL

	All	Since 2014
Citations	261500	151736
h-index	142	106
i10-index	338	234

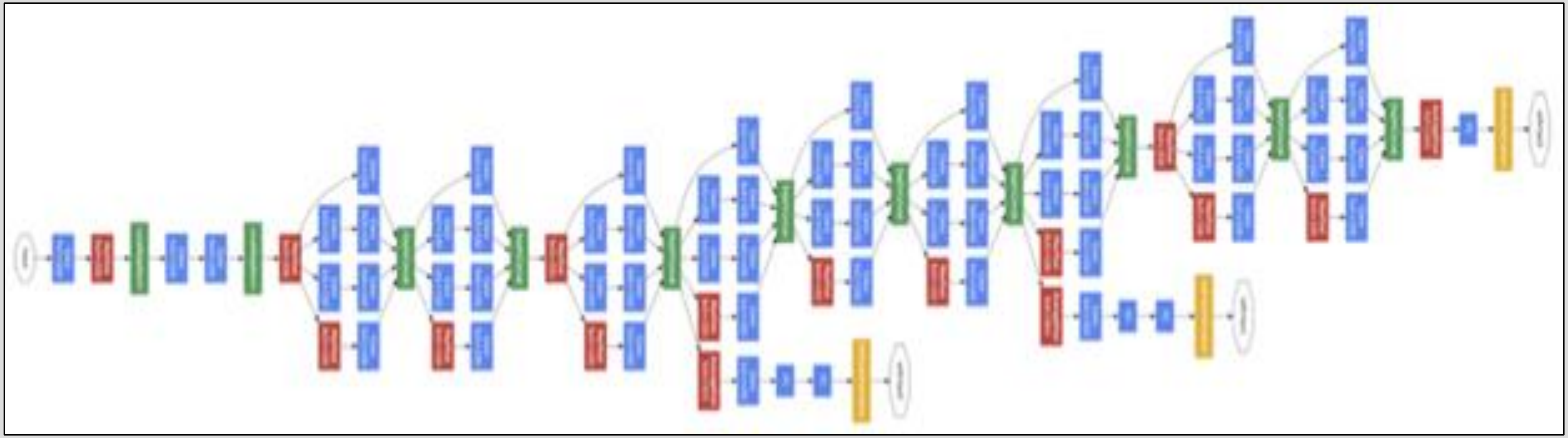
Deep Neural Networks

- Not all deep learning is NN
- DNN are many layers
- Deep Convolutional NN (DCNN) most common
  - AlexNet (1970s resurrected)



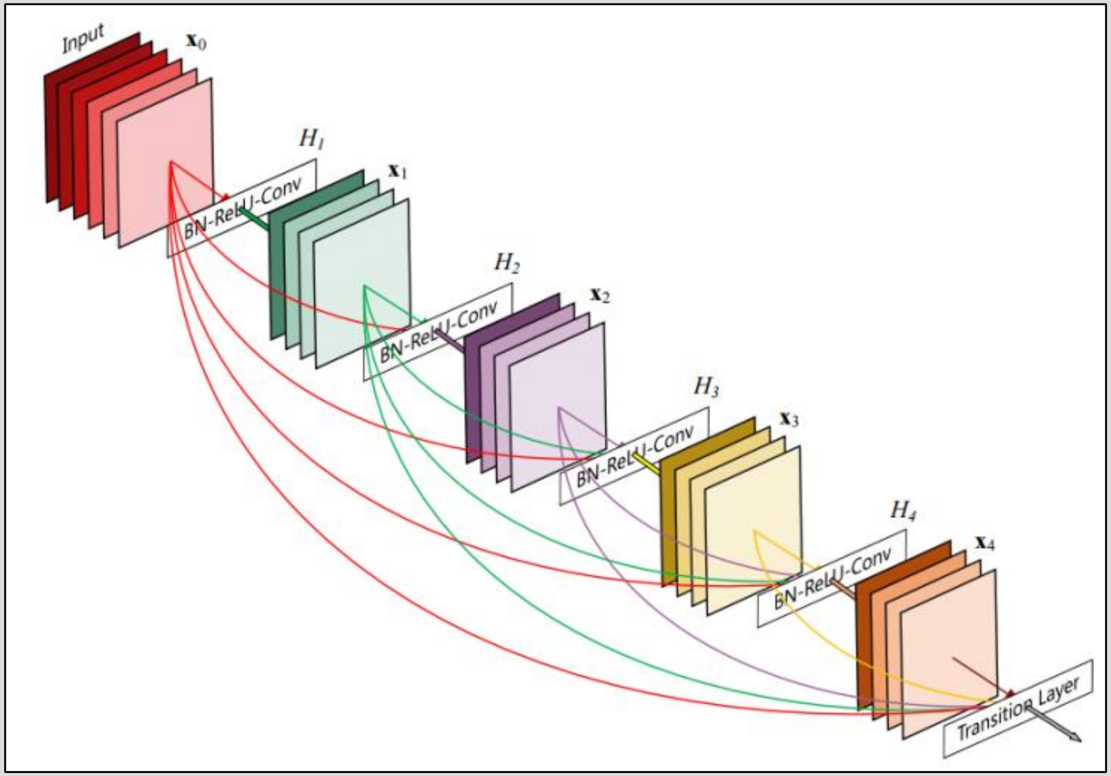
Deep Neural Networks

- Not all deep learning is NN
- DNN are many layers
- Deep Convolutional NN (DCNN) most common
  - AlexNet (1970s resurrected)
  - Inception Networks



Deep Neural Networks

- Not all deep learning is NN
- DNN are many layers
- Deep Convolutional NN (DCNN) most common
  - AlexNet
  - Inception Networks
  - Residual Networks

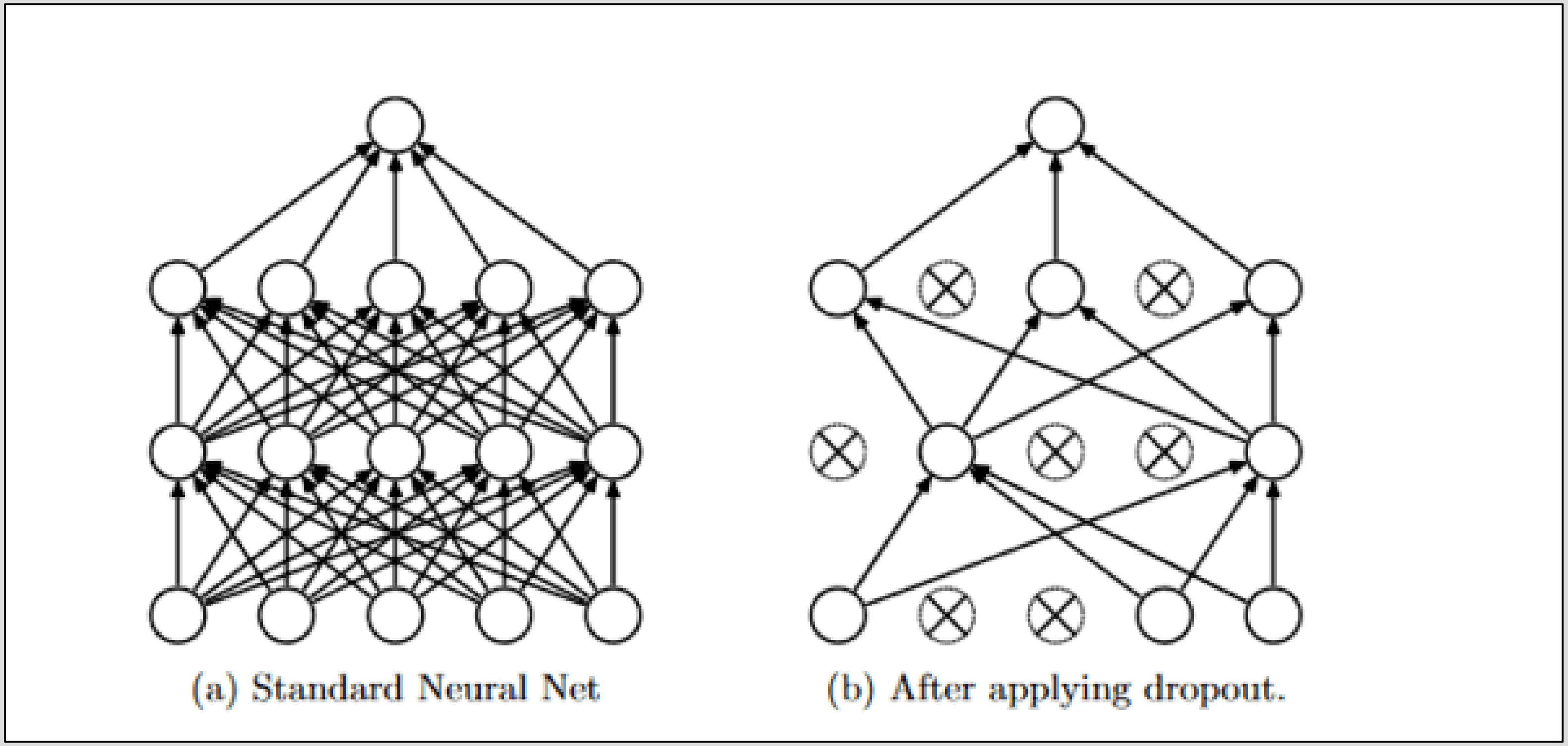


Different Parts

- Dropout
- Nonlinearity
- Pooling
- Batch normalization

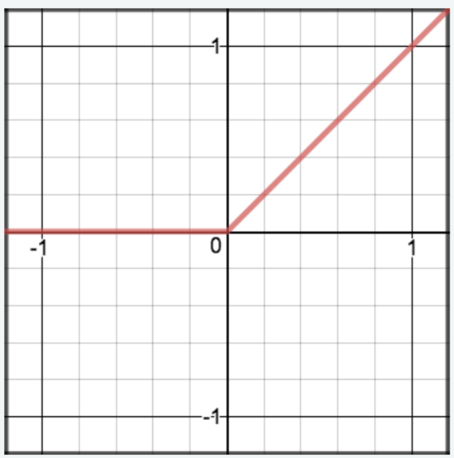
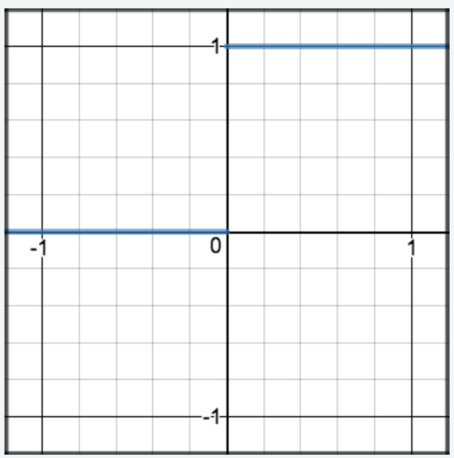
Dropout

- Drop neurons at random during training
- Try to avoid overfitting

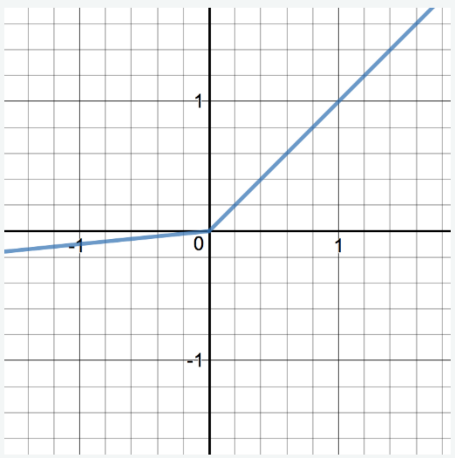
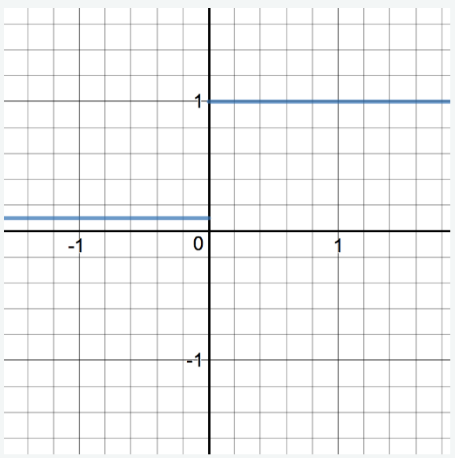


https://www.commonlounge.com/discussion/694fd08c36994186a48d122e511f29d5

Nonlinearity

Function	Derivative
$R(z) = \begin{Bmatrix} z & z > 0 \\ 0 & z <= 0 \end{Bmatrix}$	$R'(z) = \begin{Bmatrix} 1 & z > 0 \\ 0 & z < 0 \end{Bmatrix}$
	

Rectified Linear Units

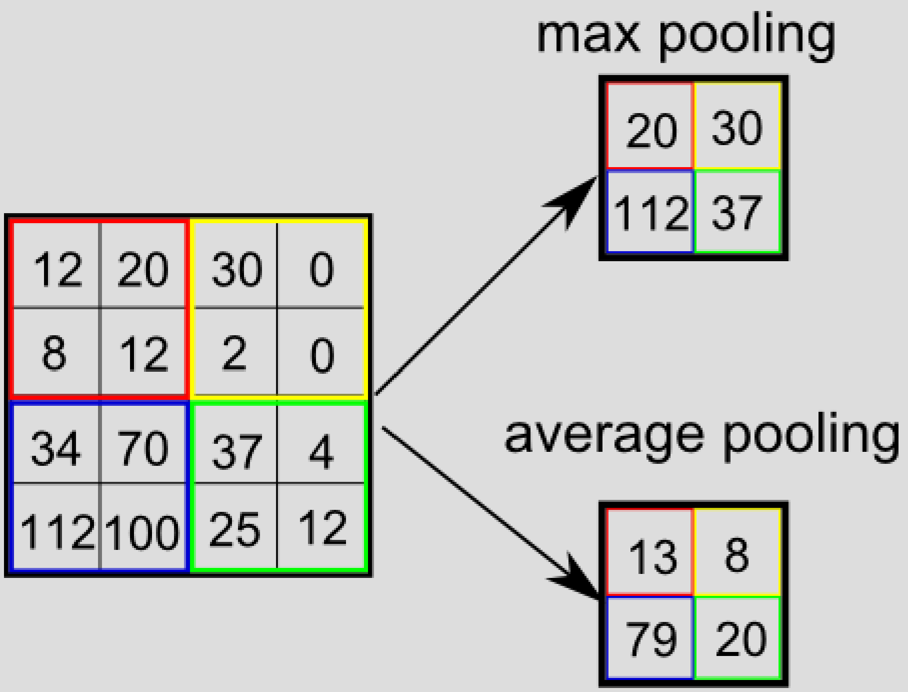
Function	Derivative
$R(z) = \begin{Bmatrix} z & z > 0 \\ \alpha z & z <= 0 \end{Bmatrix}$	$R'(z) = \begin{Bmatrix} 1 & z > 0 \\ \alpha & z < 0 \end{Bmatrix}$
	

Leaky RELU

https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html

Pooling

- Helps address scale, noise, overfitting, etc.



http://cs231n.github.io/convolutional-networks/
https://medium.com/data-science-group-iitr/building-a-convolutional-neural-network-in-python-with-tensorflow-d251c3ca8117

Batch Normalization

- Internal covariate shift
  - Distribution of activations is constantly changing during training
  - Slows down learning
- Do in training
- Remove for testing
- Run before nonlinearity

https://towardsdatascience.com/batch-normalization-theory-and-how-to-use-it-with-tensorflow-1892ca0173ad

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

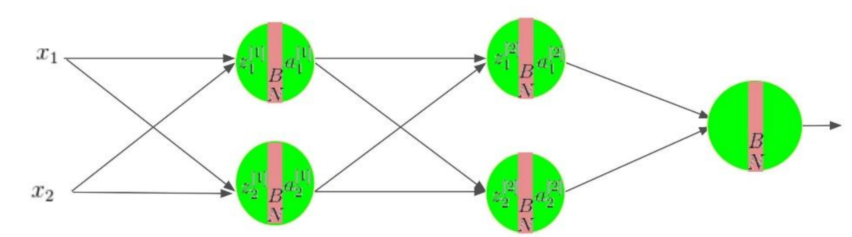
$$\mu_\mathcal{B} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

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

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

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

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

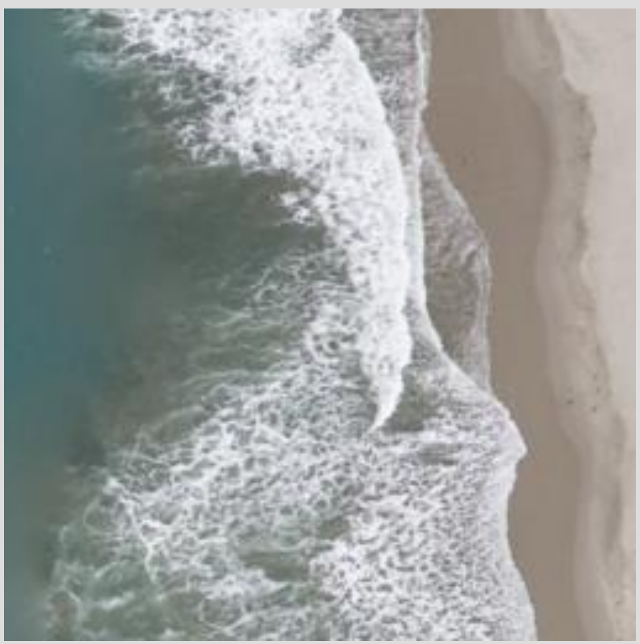


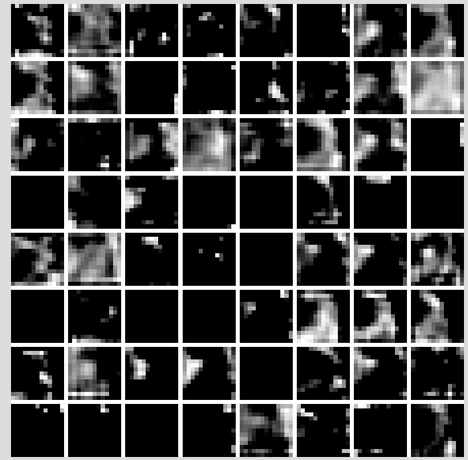
$$z^{[l]} = W^{[l]}a^{[l-1]} \longrightarrow \begin{aligned} \mu^{[l]} &= \frac{1}{m}\sum_i z^{[l](i)} \\ \sigma^{[l]2} &= \frac{1}{m}\sum_i (z^{[l](i)} - \mu^{[l]})^2 \\ z_{norm}^{[l](i)} &= \frac{z^{[l](i)} - \mu^{[l]}}{\sqrt{\sigma^{[l]2} + \epsilon}} \\ \tilde{z}^{[l](i)} &= \gamma^{[l]} z_{norm}^{[l](i)} + \beta^{[l]} \end{aligned} \longrightarrow a^{[l]} = g^{[l]}(\tilde{z}^{[l]})$$

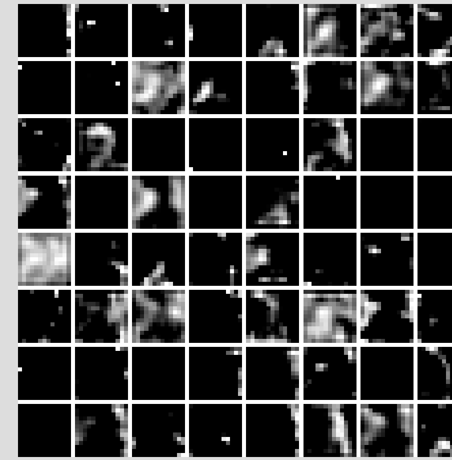
DATA SCIENCE
& ANALYTICS

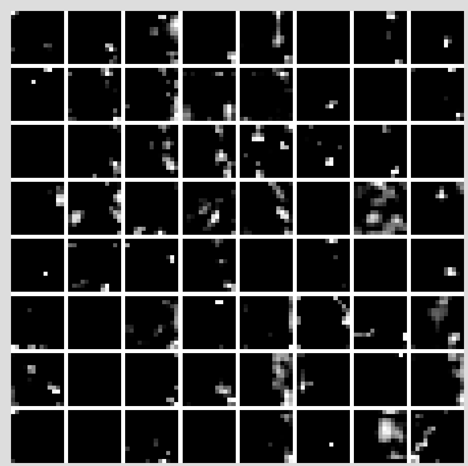
Deep Convolutional NN

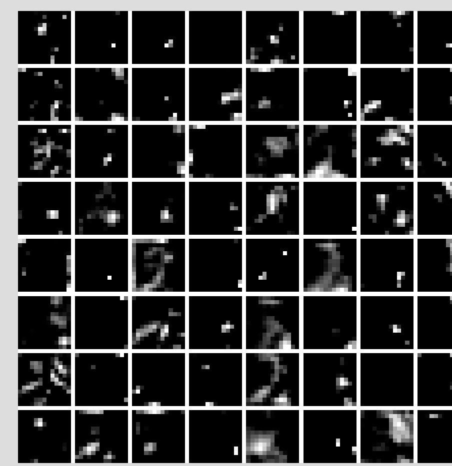
- Create layers of feature maps from kernels







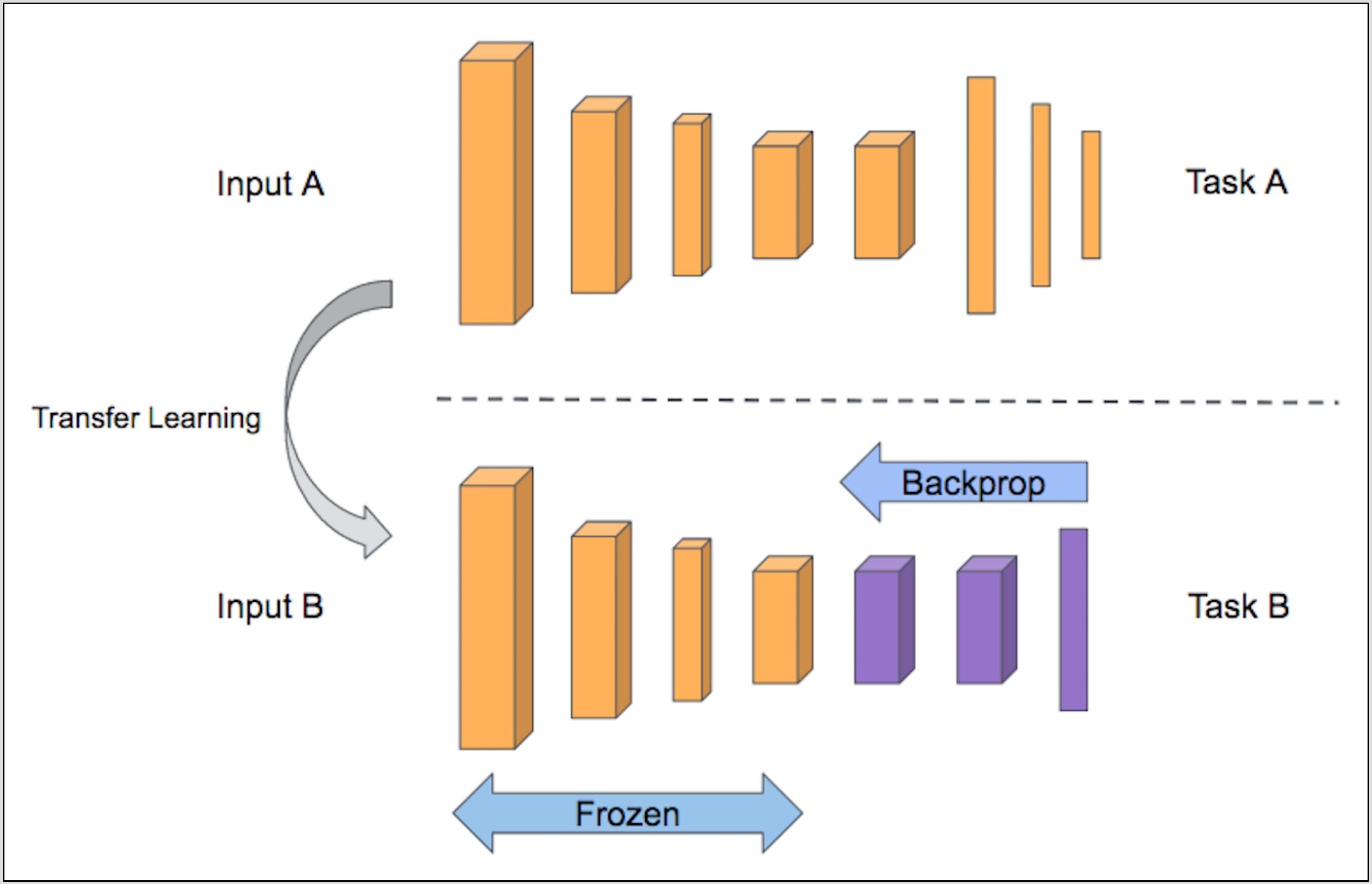




Deep Learning

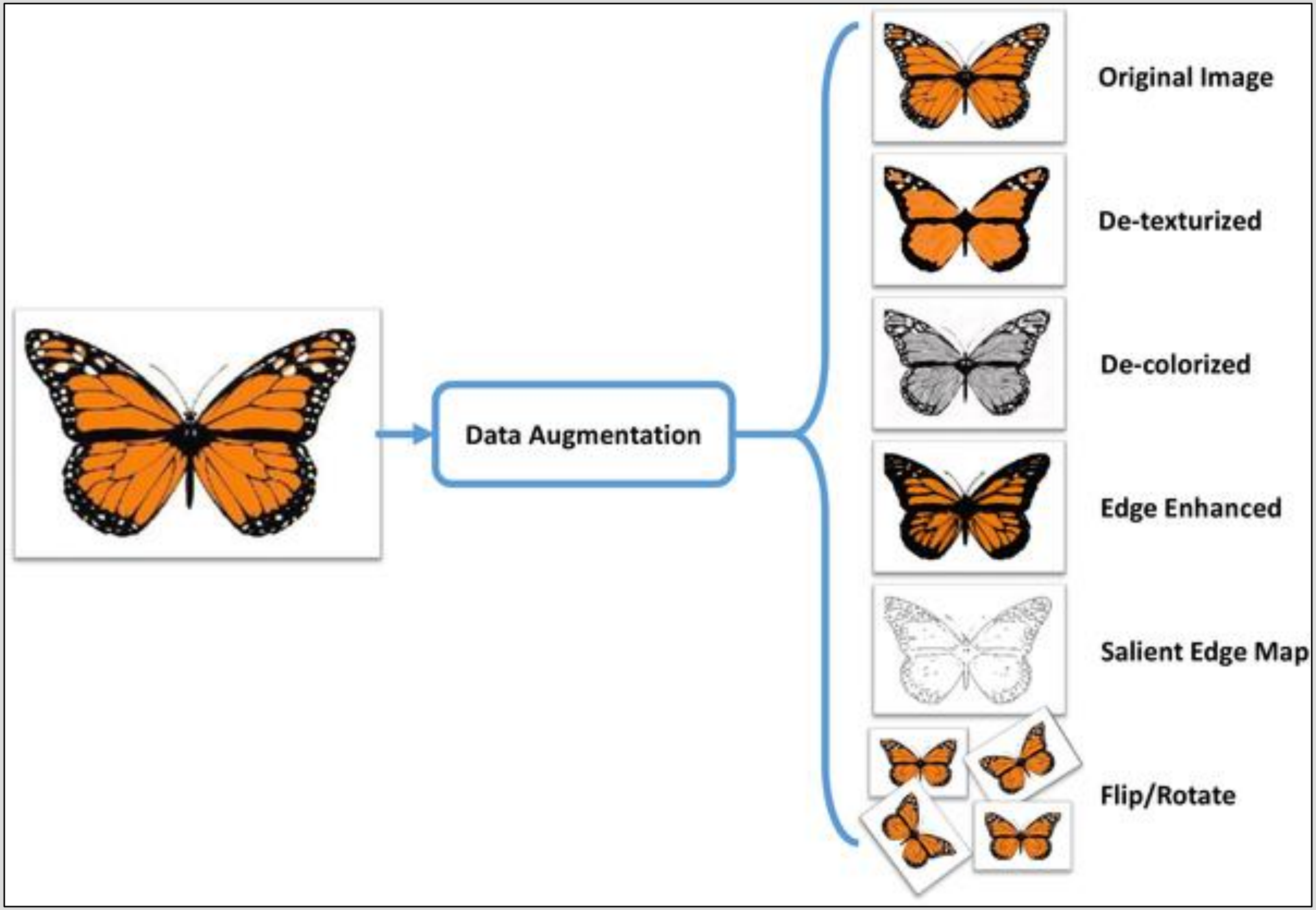
- Typically many many layers/steps of processing
- Learning multiple levels of feature extraction
  - Shallow levels, simple features
  - Working up to more abstract features (concepts)
- https://en.wikipedia.org/wiki/Deep_learning

Transfer Learning



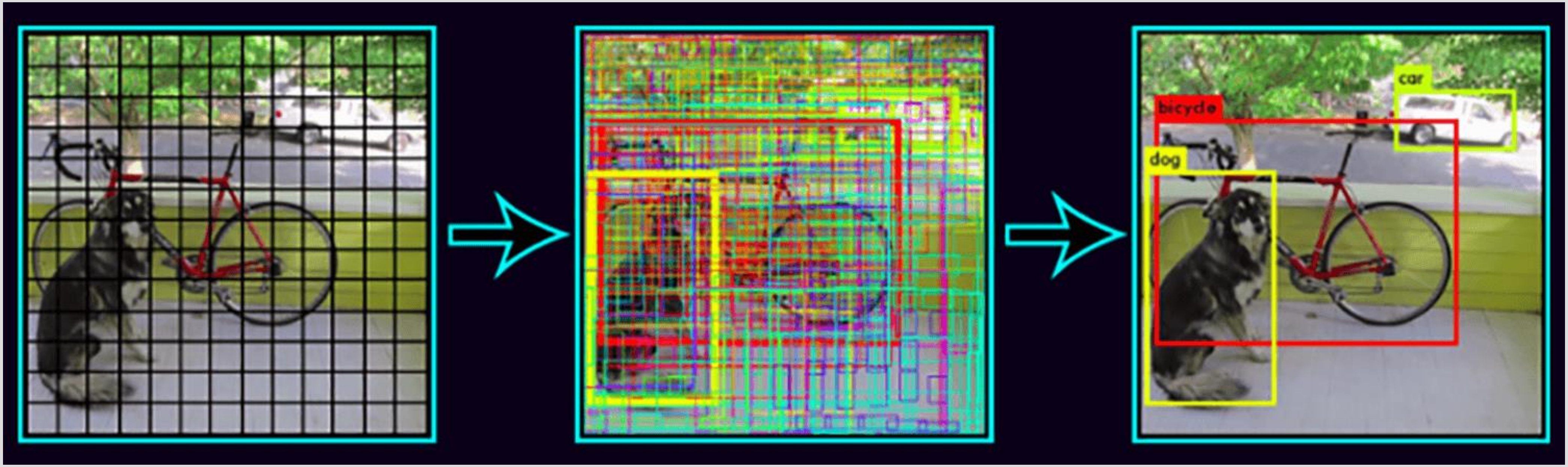
https://medium.com/@subodh.malgonde/transfer-learning-using-tensorflow-52a4f6bcde3e

Data Augmentation



https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0183838

Localization: YOLO and RCNNs and …

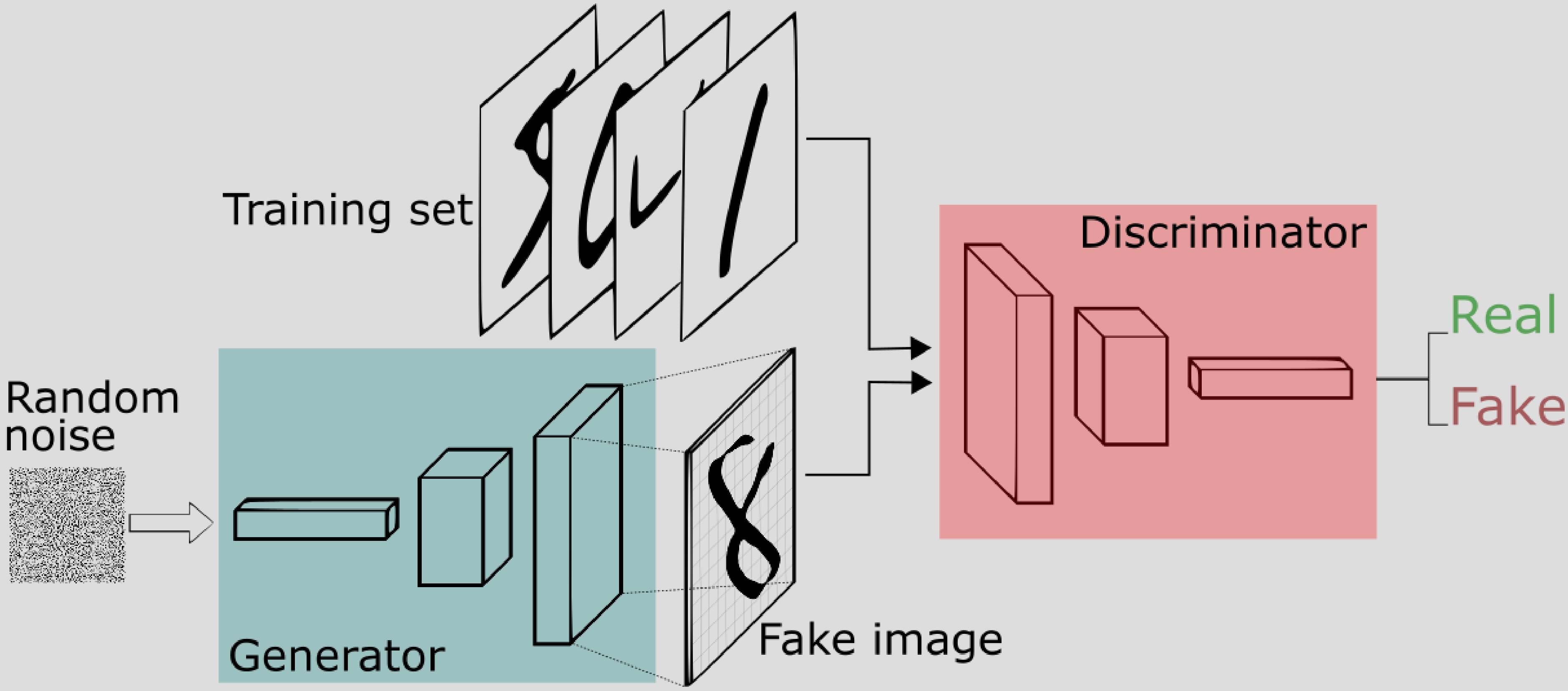


Divide into regions

Detection

Final decisions

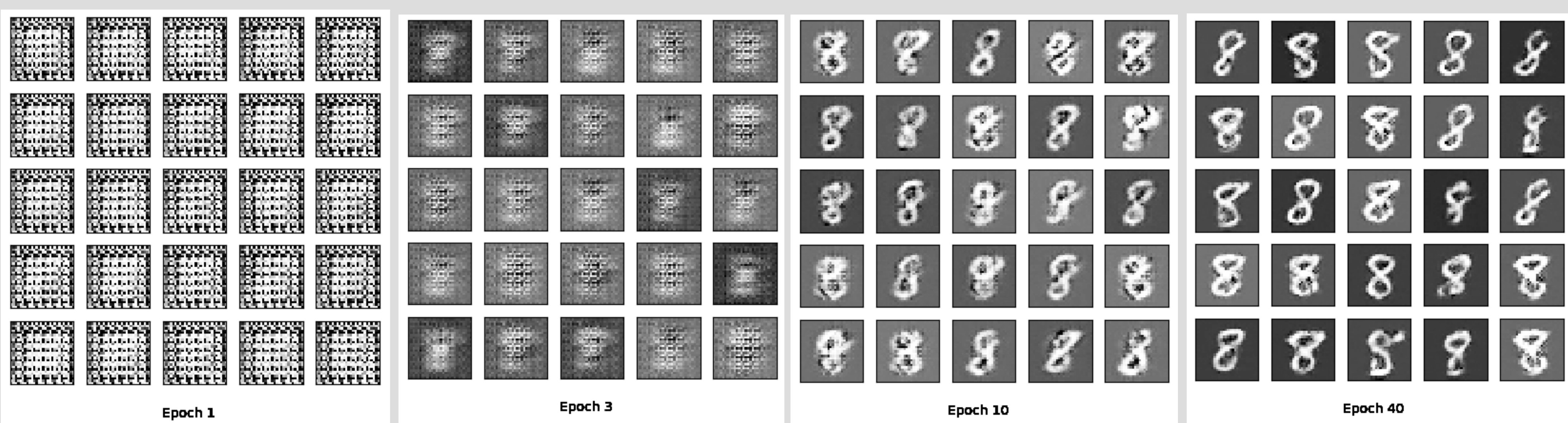
Generative Adversarial Networks



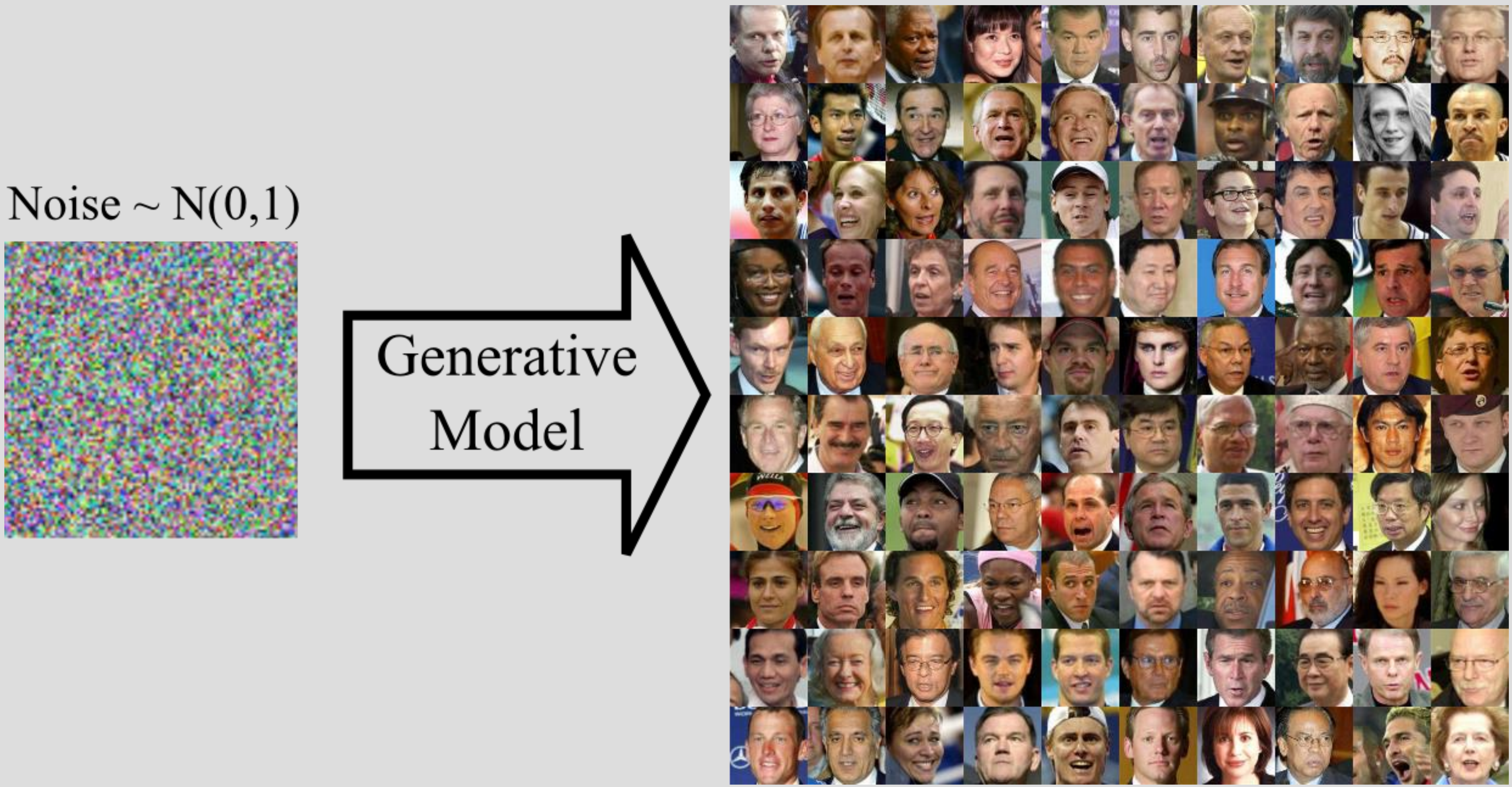
https://skymind.ai/wiki/generative-adversarial-network-gan

DATA SCIENCE
& ANALYTICS

GAN on MINST



Generative Adversarial Networks



http://torch.ch/blog/2015/11/13/gan.html

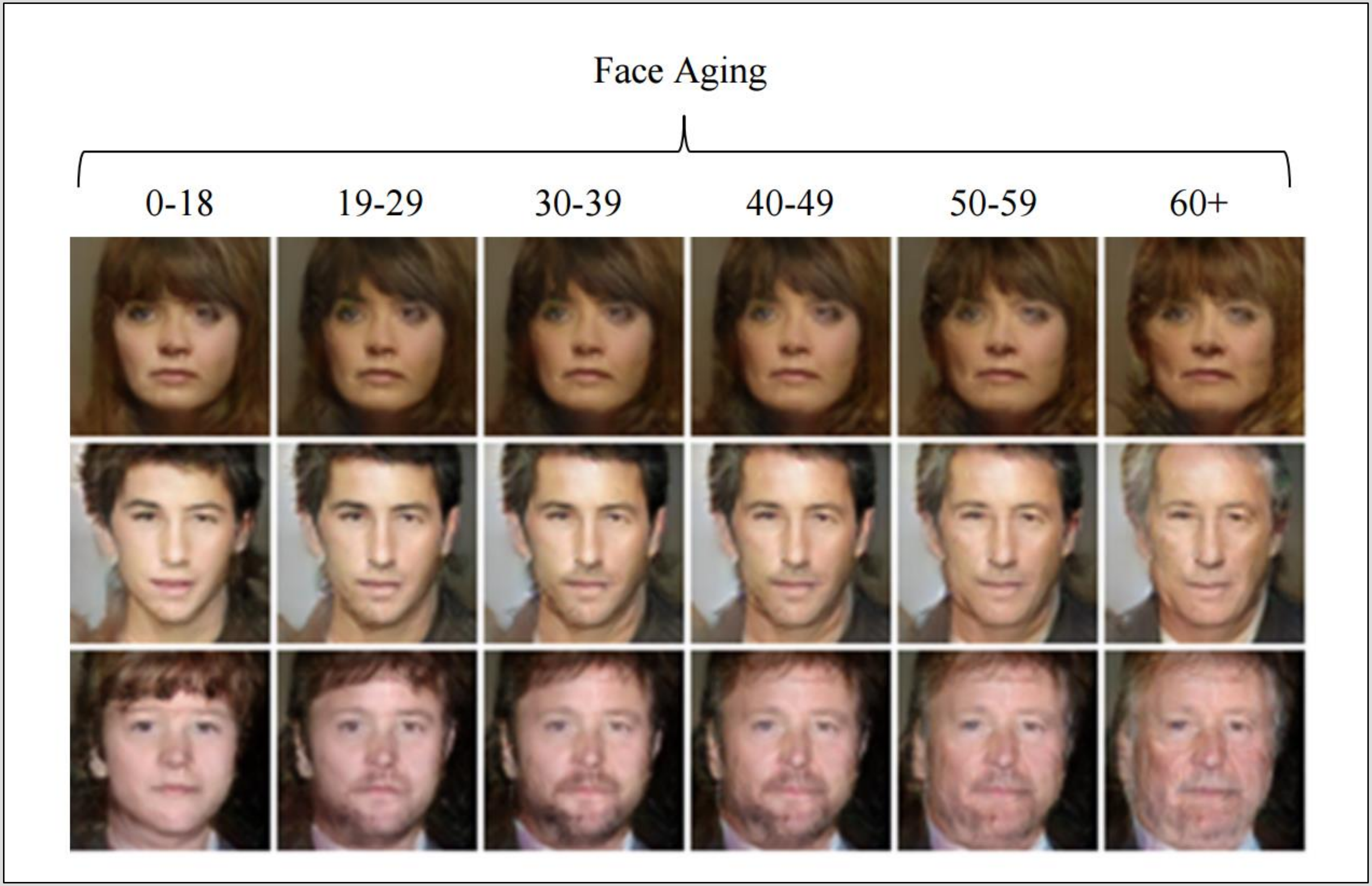
Generative Adversarial Networks

after 5 epochs

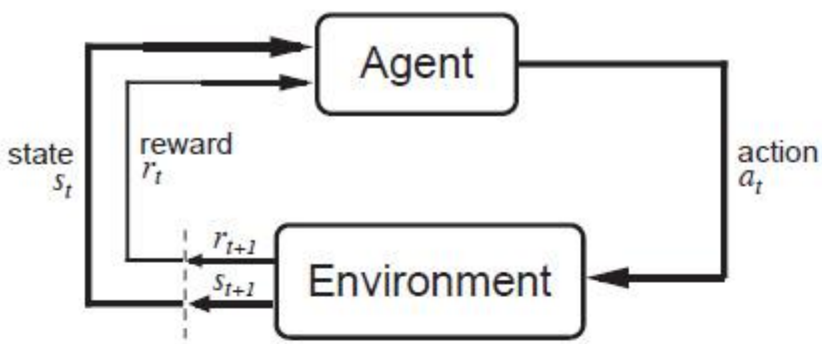
after 100 epochs

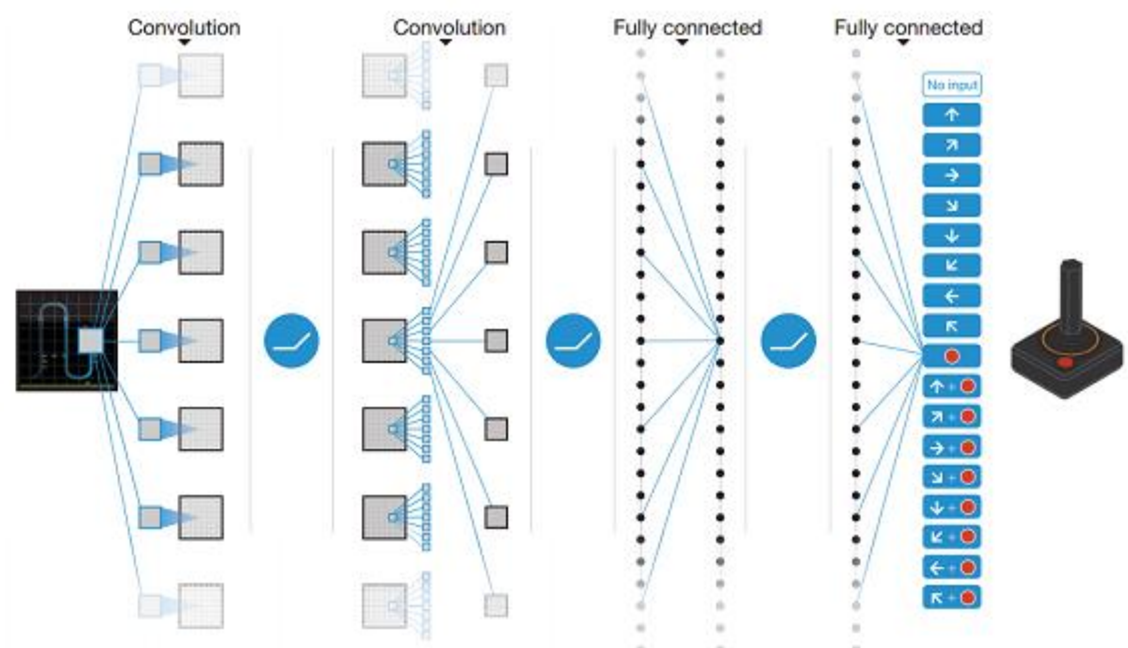
http://torch.ch/blog/2015/11/13/gan.html

Generative Adversarial Networks



Deep Reinforcement Learning





https://adeshpande3.github.io/Deep-Learning-Research-Review-Week-2-Reinforcement-Learning

Questions?