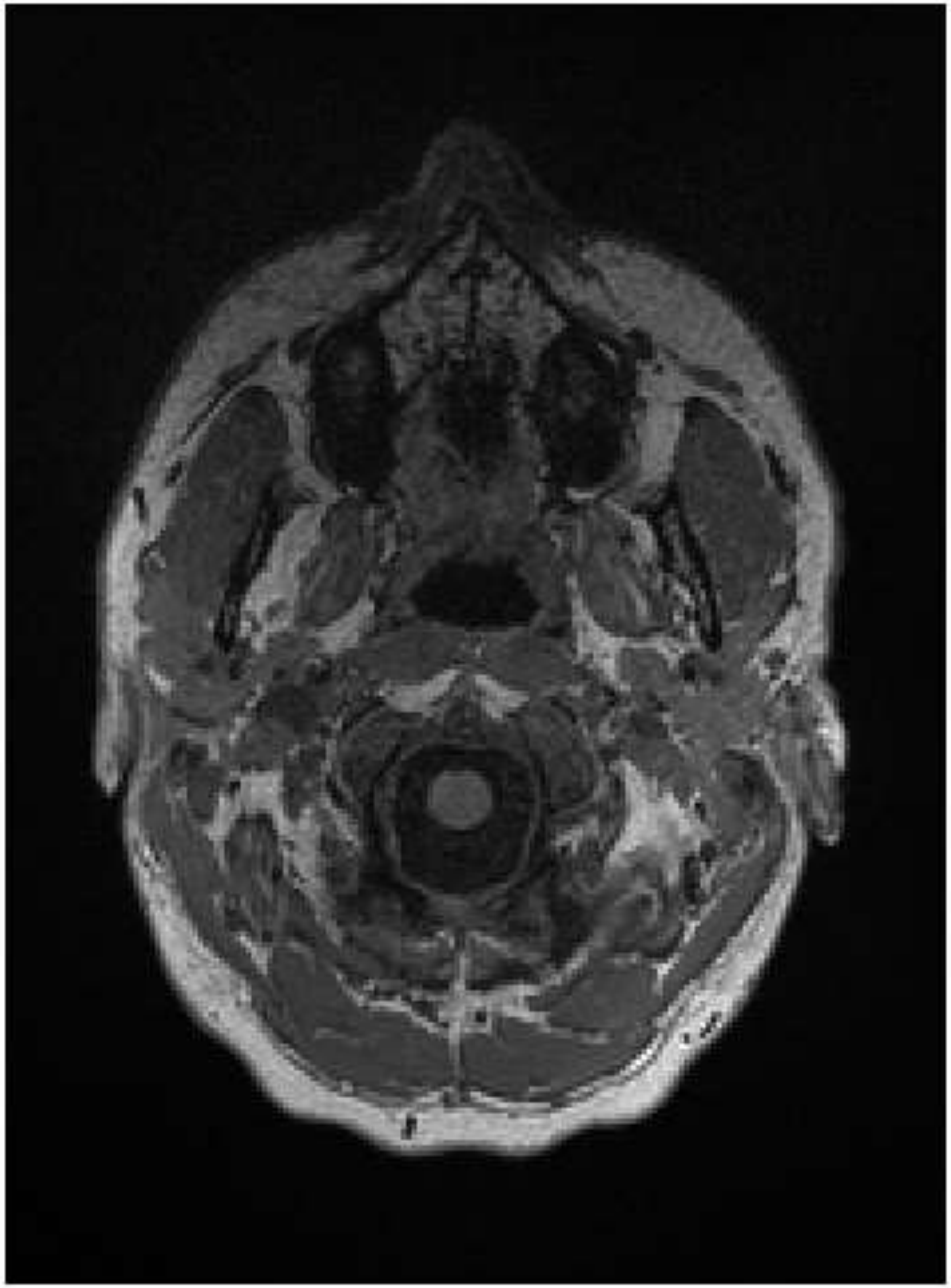
Intro to Deep Learning for NeuroImaging

Andrew Doyle

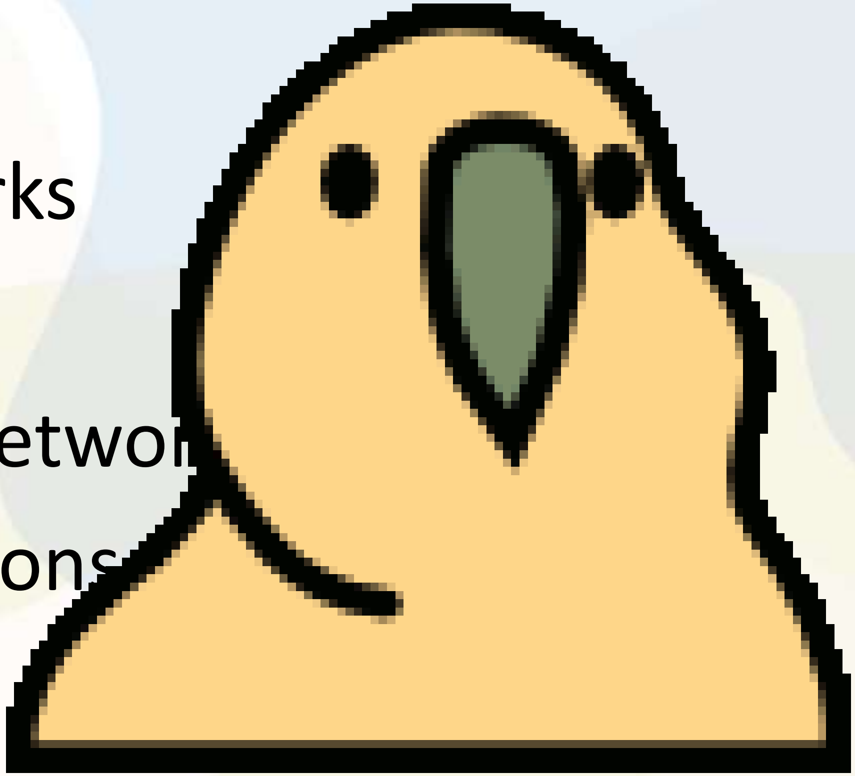
McGill Centre for Integrative Neuroscience

@crocodoyle

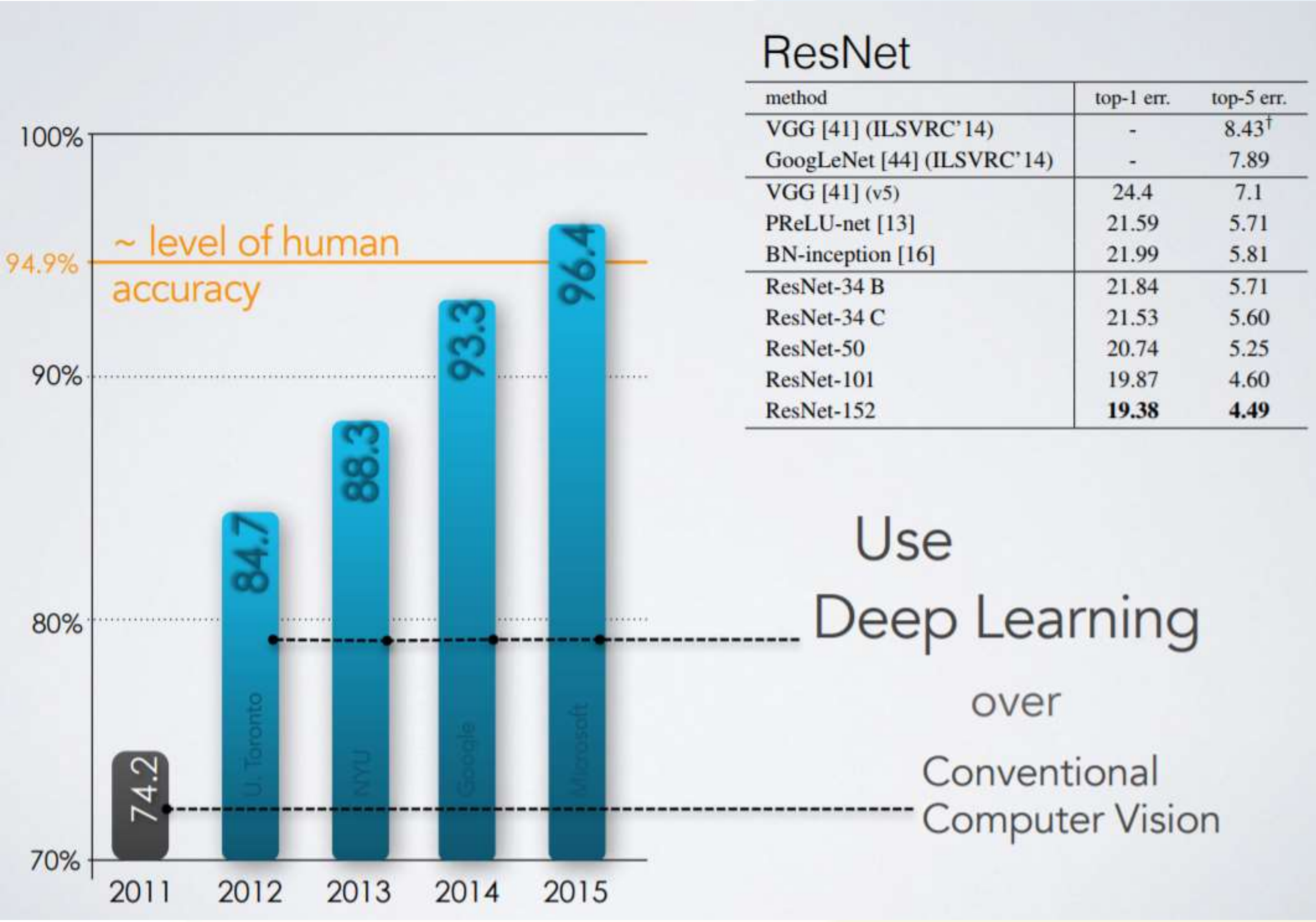


Outline

1. GET EXCITED
2. Artificial Neural Networks
3. Backpropagation
4. Convolutional Neural Networ
5. Neuroimaging Applications



ImageNet-1000 Results

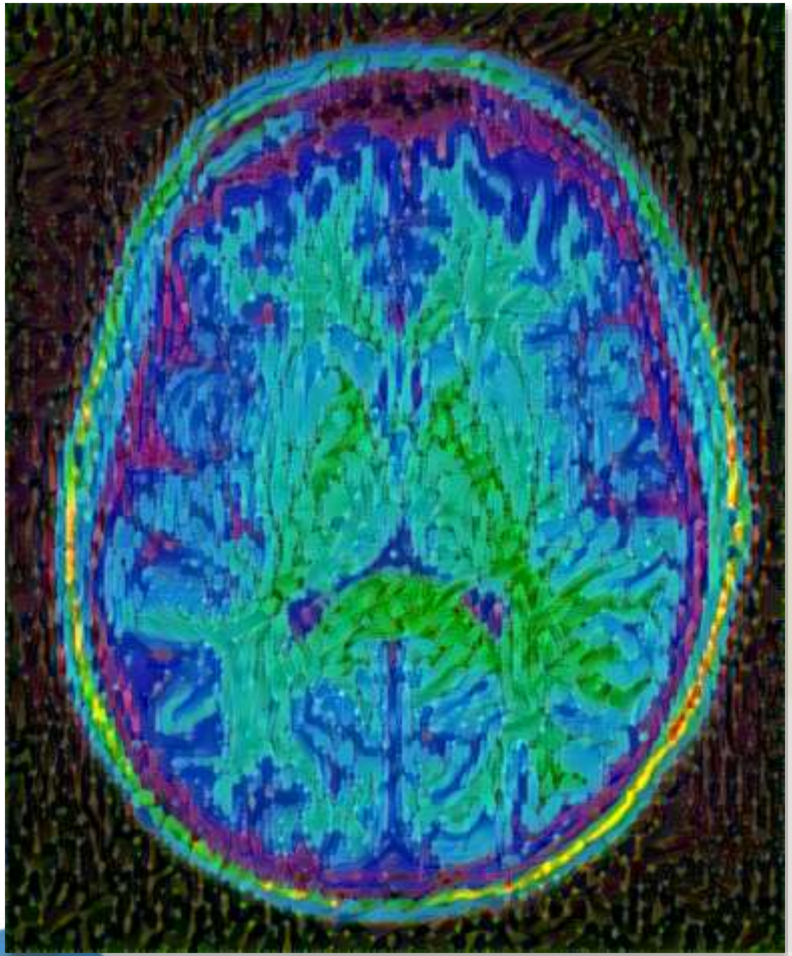


ResNet

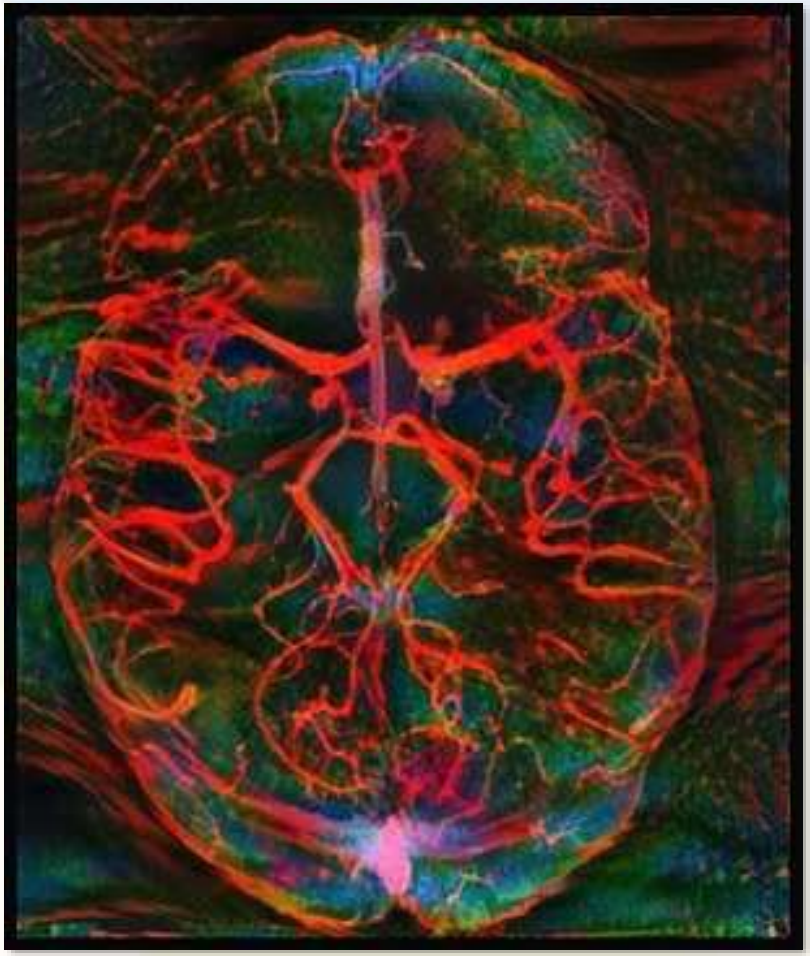
method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Image courtesy Aaron Courville, 2015

Generative Models



BrainBrush



Deep Blood by Team BloodArt

Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks." *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. IEEE, 2016.

Generative Models

StackGAN

Zhang, Han, et al. "StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks." *arXiv preprint arXiv:1612.03242* (2016).

Generative Models

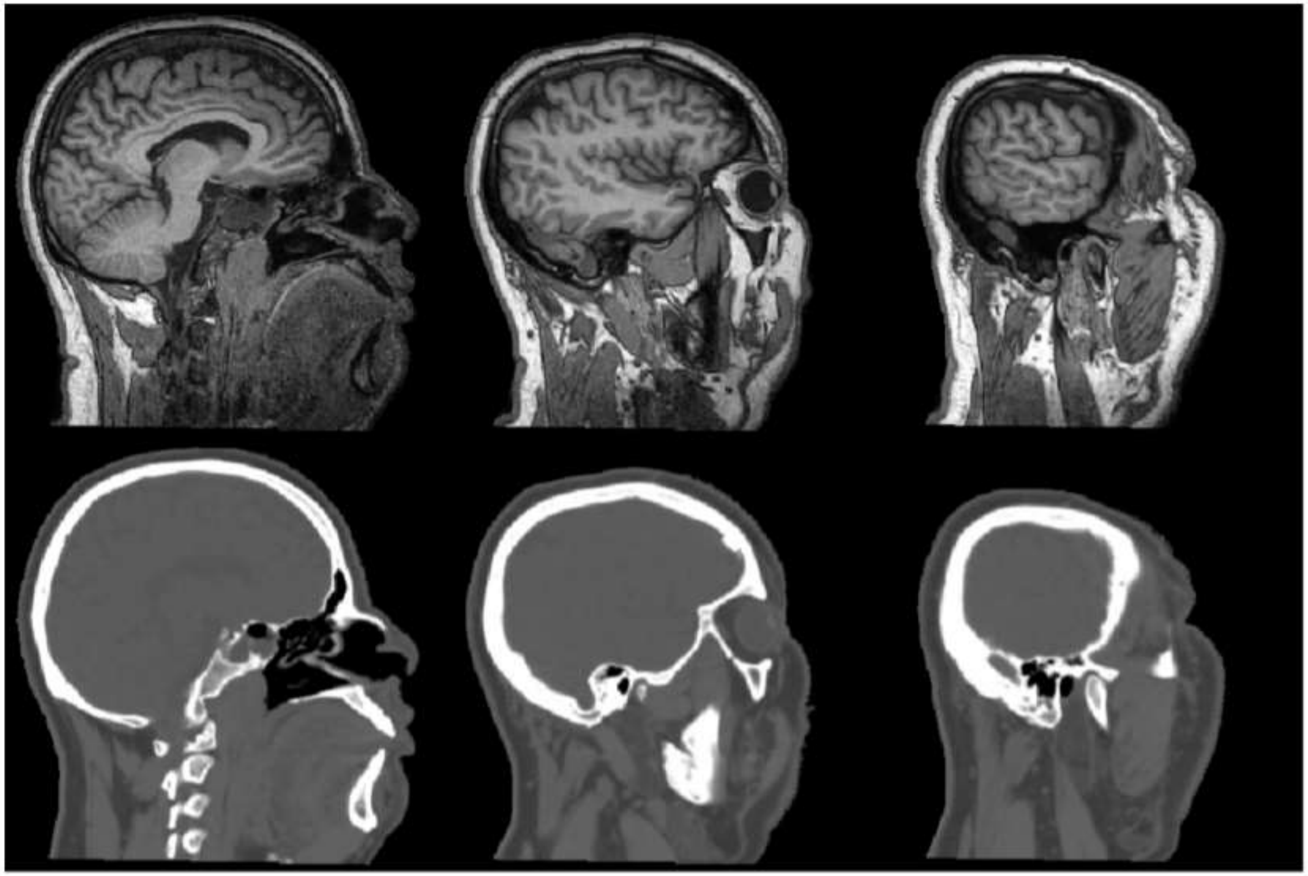
CycleGAN

Zhu, Jun-Yan, et al. "Unpaired image-to-image translation using cycle-consistent adversarial networks." *arXiv preprint arXiv:1703.10593* (2017).

Generative Models

MR

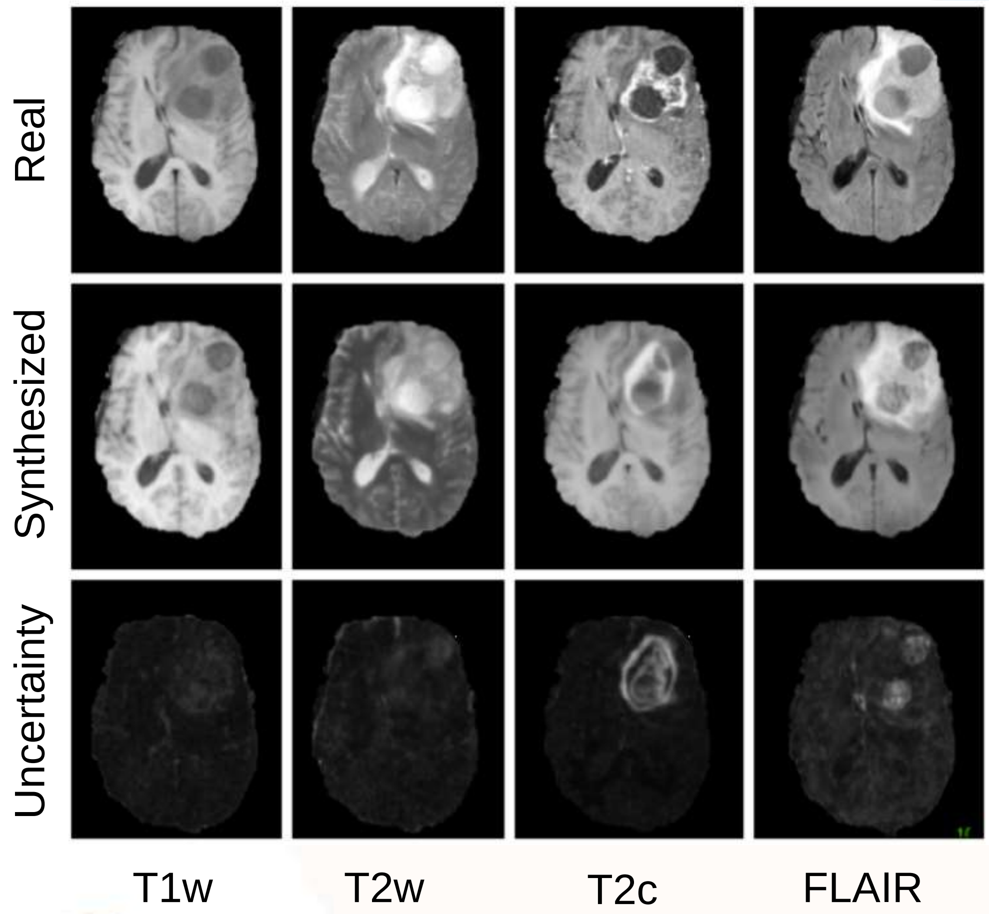
CT

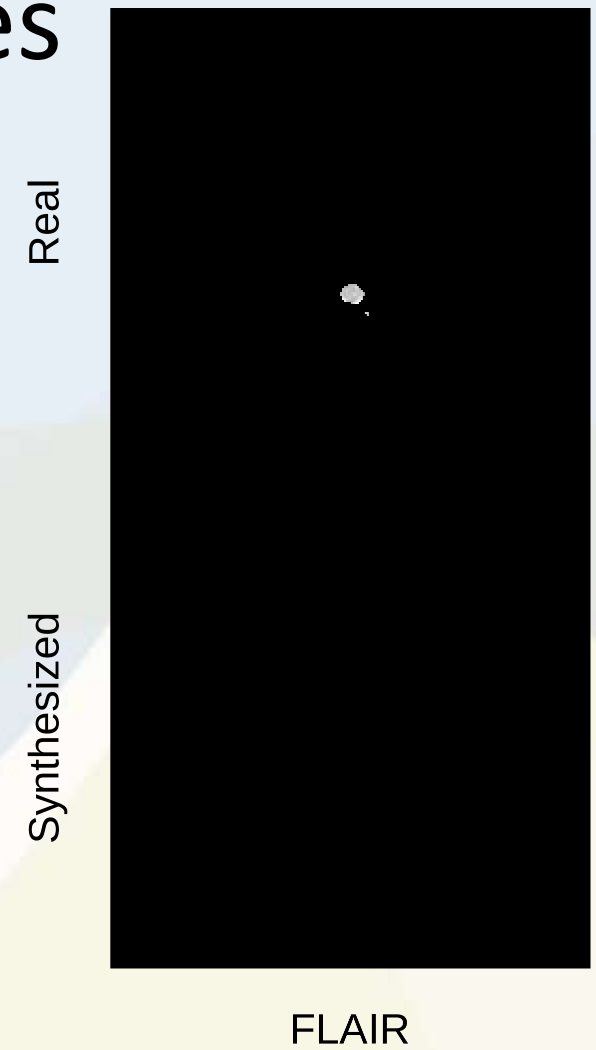


CycleGAN

Wolterink, Jelmer M., et al. "Deep MR to CT synthesis using unpaired data." *International Workshop on Simulation and Synthesis in Medical Imaging*. Springer, Cham, 2017.

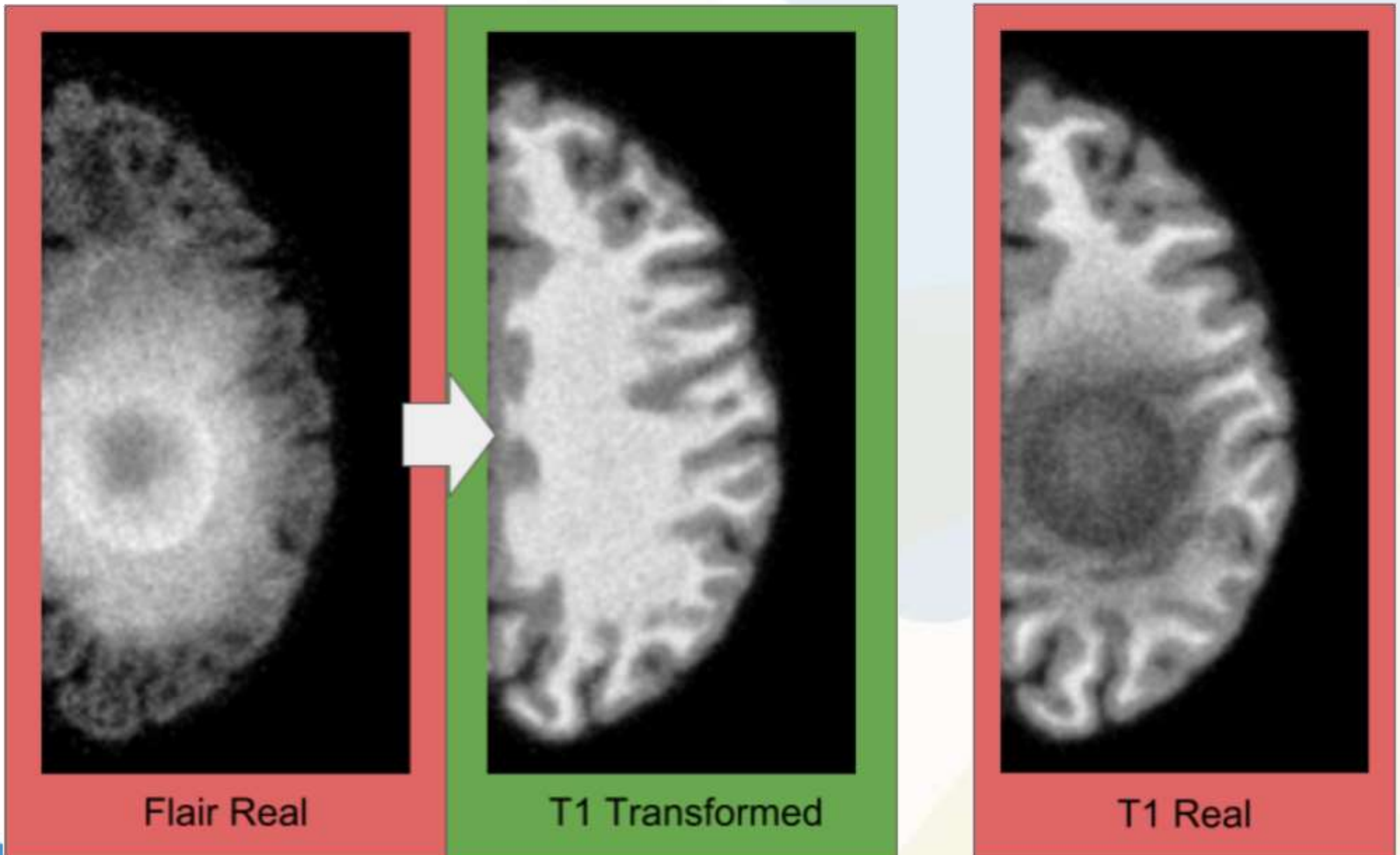
Synthetic Images





Mehta, R, et al. "RS-Net: Regression-Segmentation 3D CNN for Synthesis of Full Resolution Missing Brain MRI in the Presence of Tumours." SASHIMI Workshop of MICCAI 2018.

Synthetic Images



Cohen, Joseph Paul, Margaux Luck, and Sina Honari. "Distribution Matching Losses Can Hallucinate Features in Medical Image Translation." *arXiv preprint arXiv:1805.08841*(2018).

Reinforcement Learning

Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." *arXiv preprint arXiv:1312.5602* (2013).

Reinforcement Learning

OpenAI. “DOTA2.” August, 2017. https://blog.openai.com/dota-2/

Reinforcement Learning

mini-map

OpenAI Five Benchmark. https://blog.openai.com/openai-five-benchmark/

Montreal AI Companies

perceiv.ai

ARCTIC FOX AI

facebook research

neuro

Institut et hôpital neurologiques de Montréal
Montreal Neurological Institute and Hospital

McGill

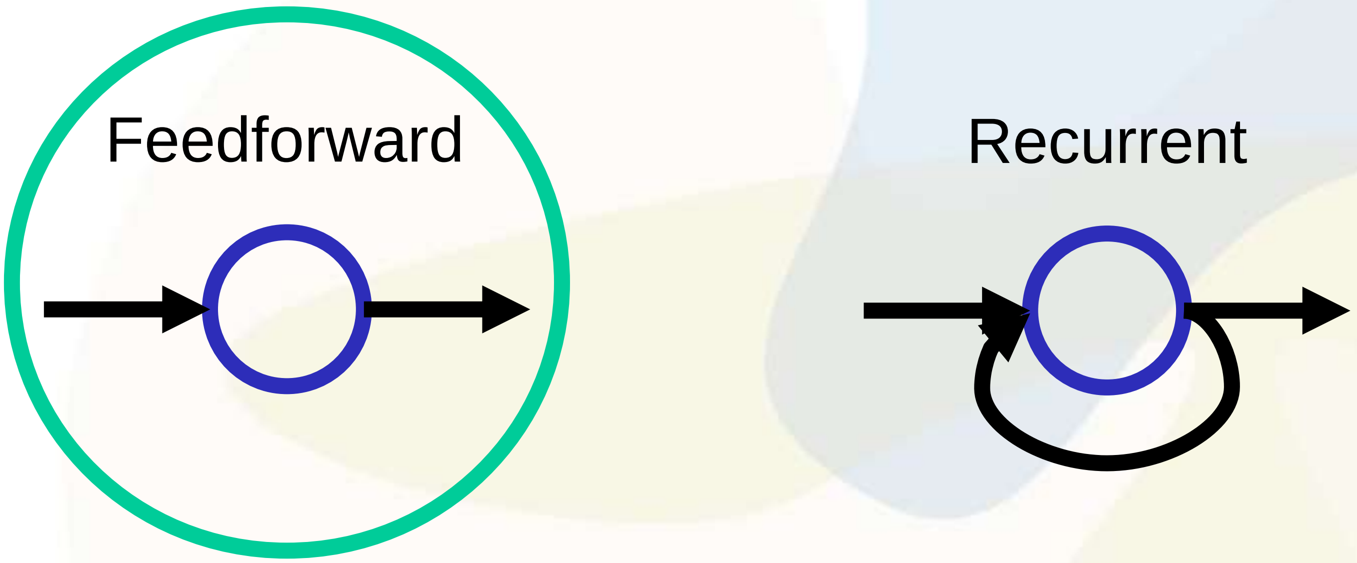
Centre universitaire de santé McGill
McGill University Health Centre

Deep Learning

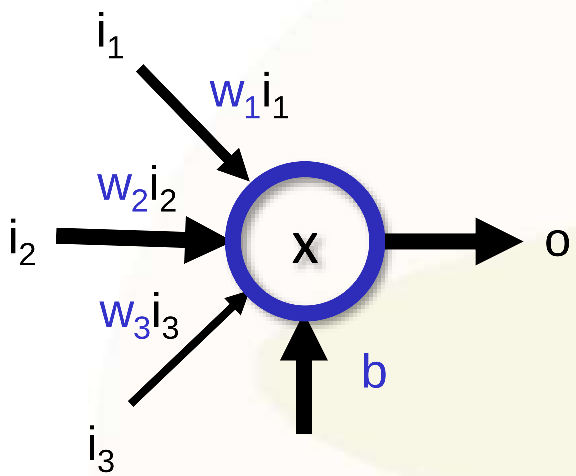
For Deep Learning, you need:

1. Artificial Neural Network
2. Loss
3. Optimizer
4. Data

Artificial Neurons

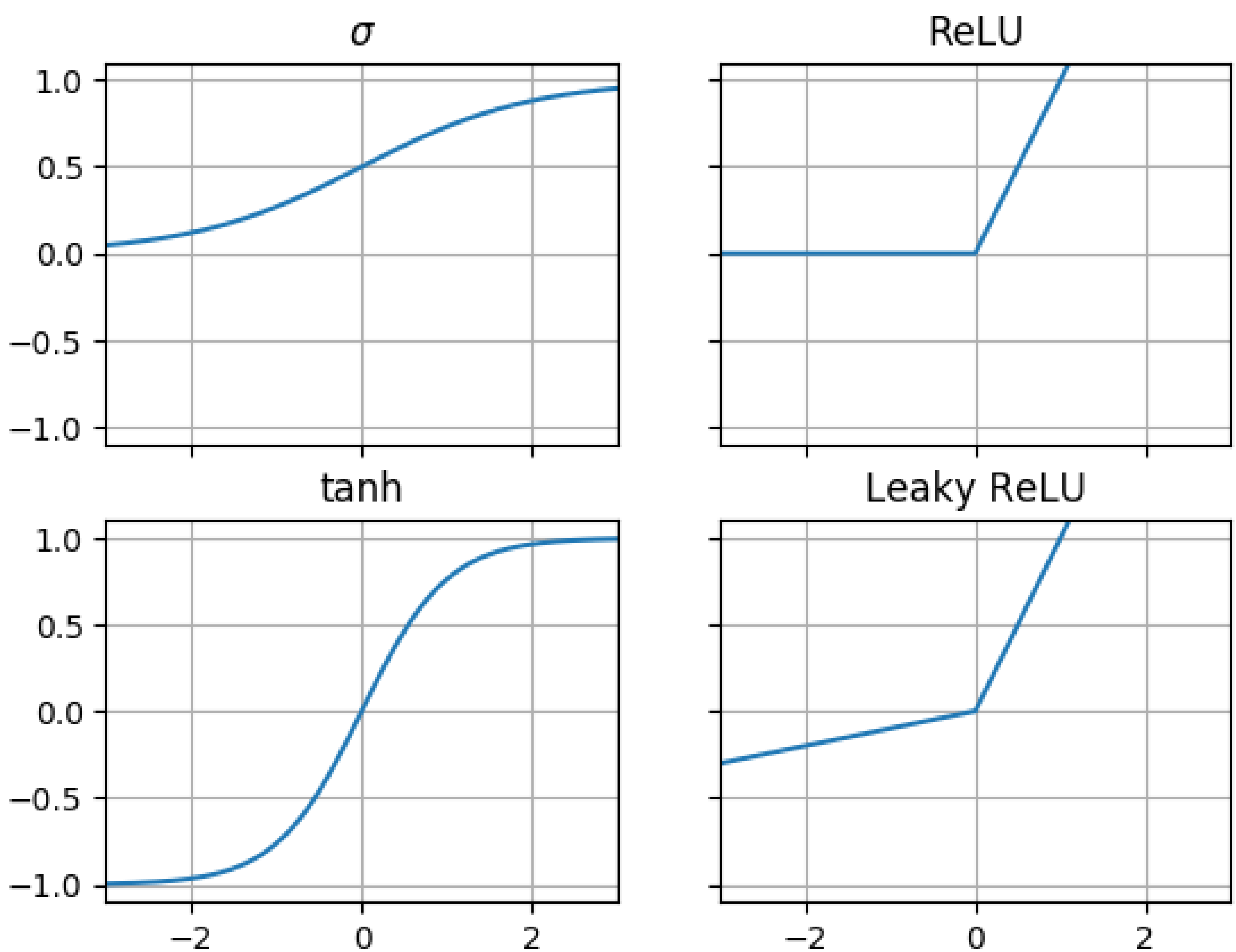


Artificial Neurons

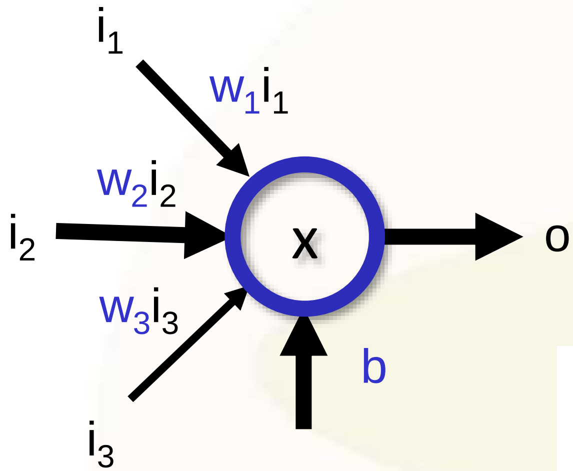


$$o = f(x) = f\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{i} + \boldsymbol{b}\right)$$

Artificial Neurons

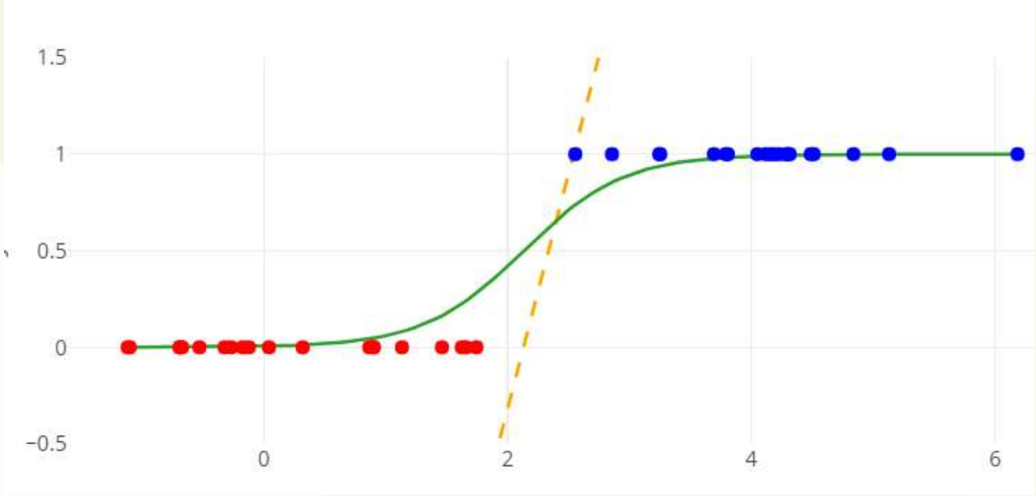


Artificial Neurons



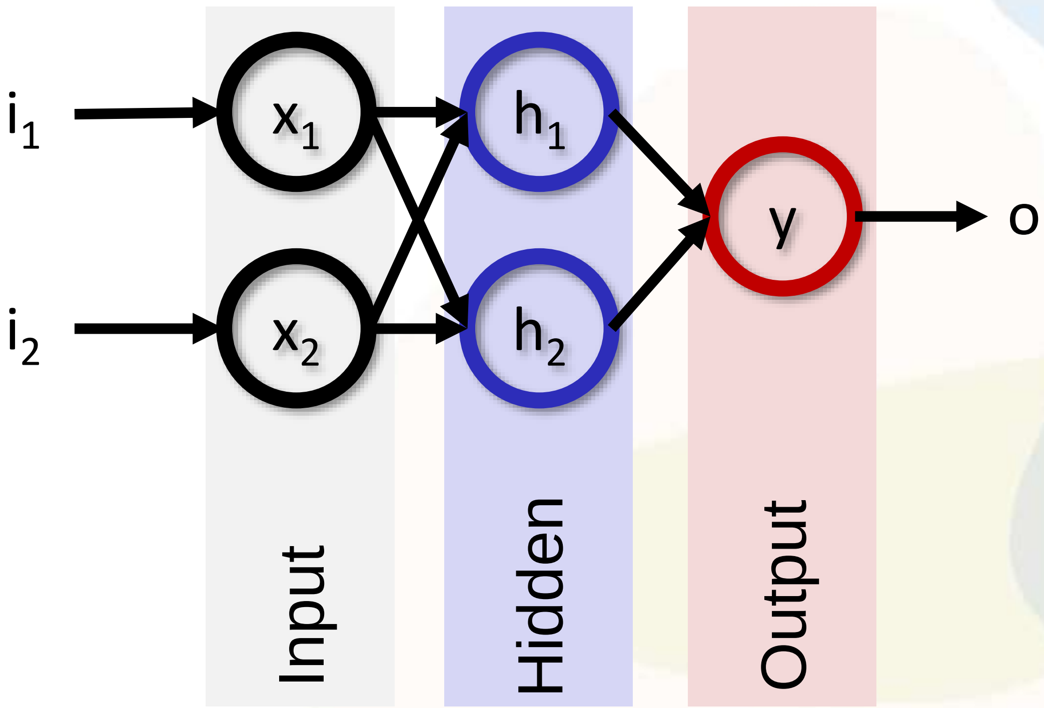
Logistic Regression

$$o = \sigma(x) = \sigma\left(\boldsymbol{w}^{\boldsymbol{T}}\boldsymbol{i} + \boldsymbol{b}\right)$$

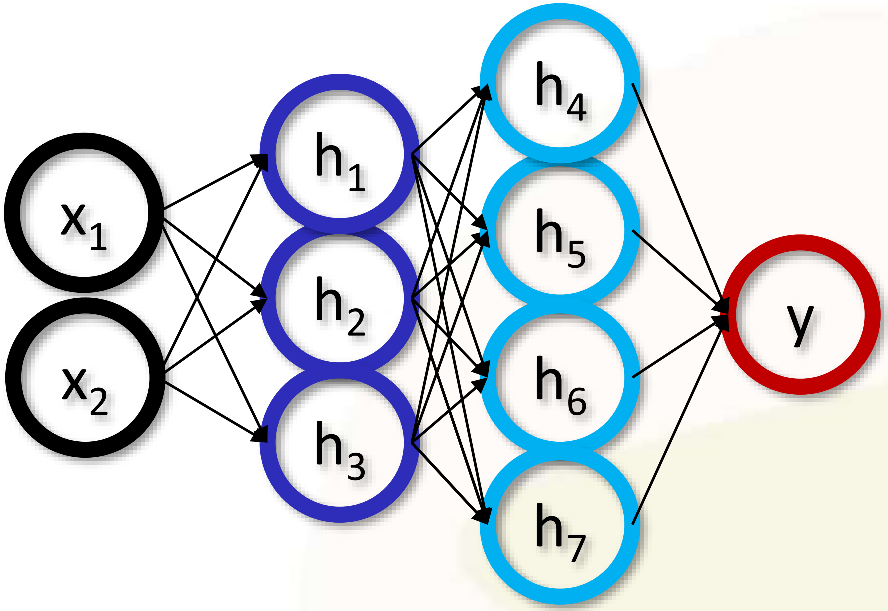


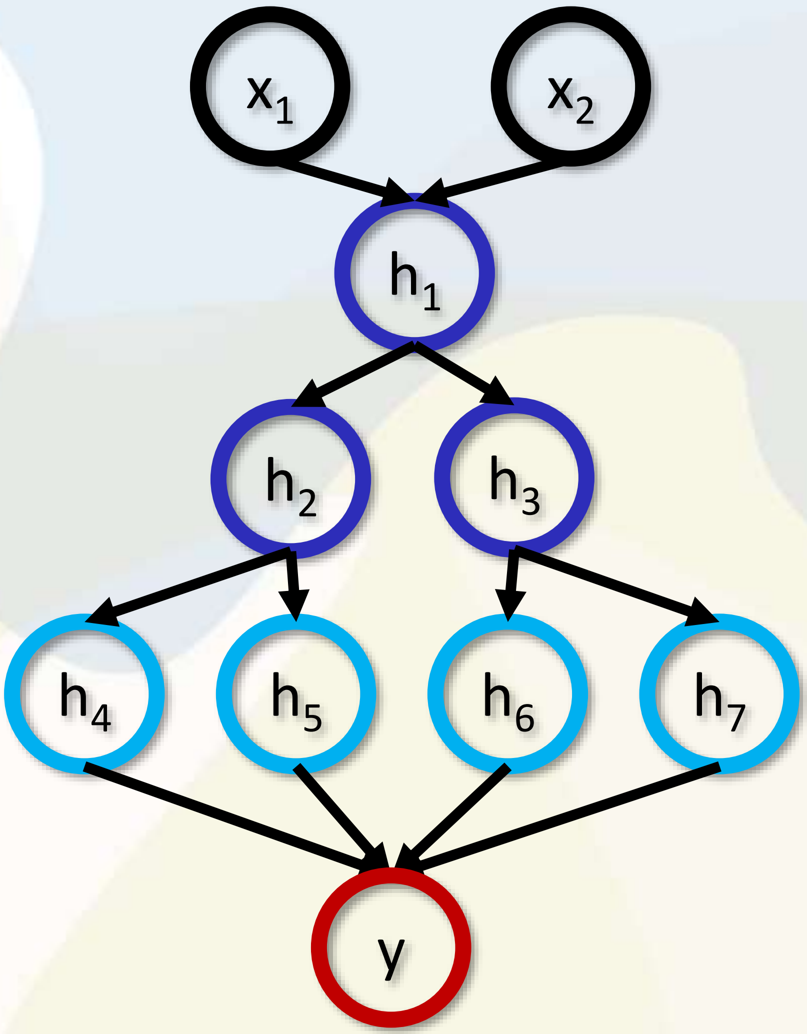
Hartl, Florian. "Logistic Regression – Geometric Intuition"
https://florianhartl.com/logistic-regression-geometric-intuition.html

Neural Networks



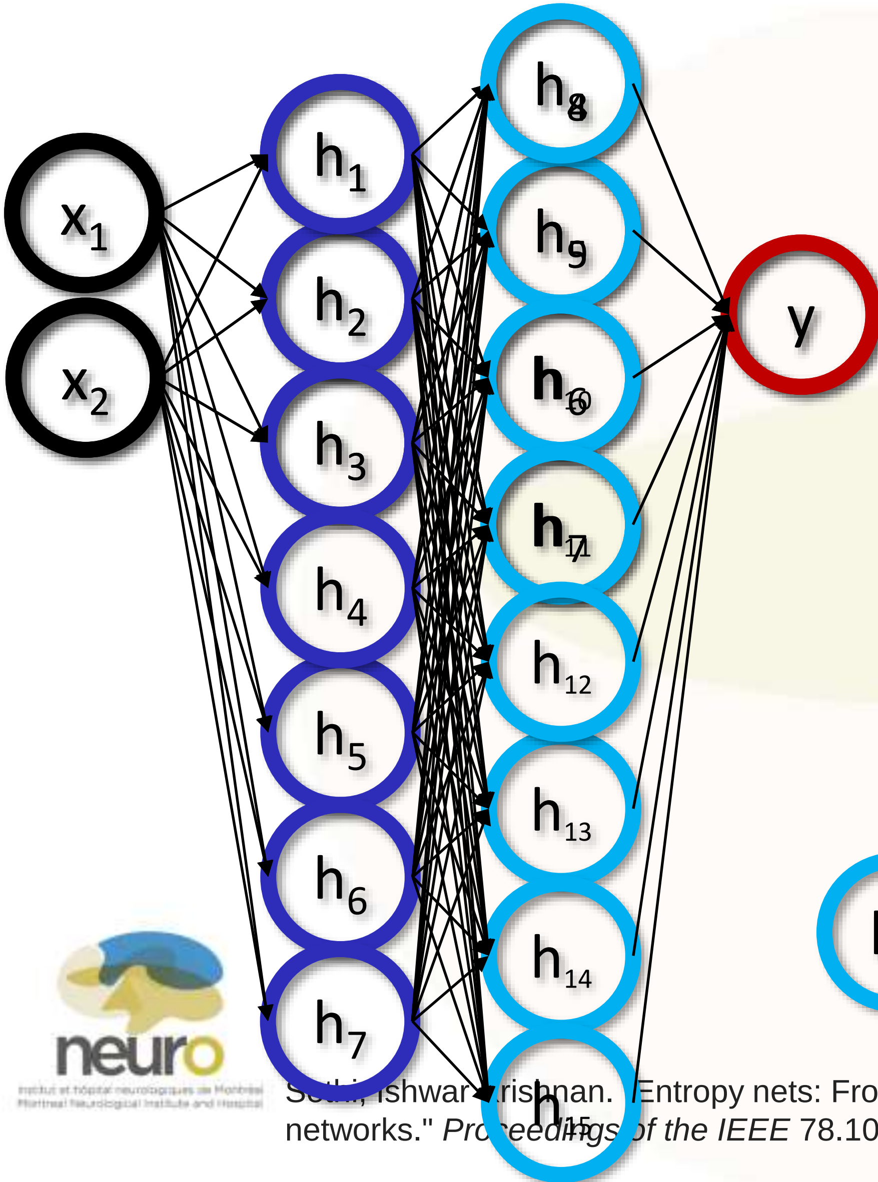
Neural Networks

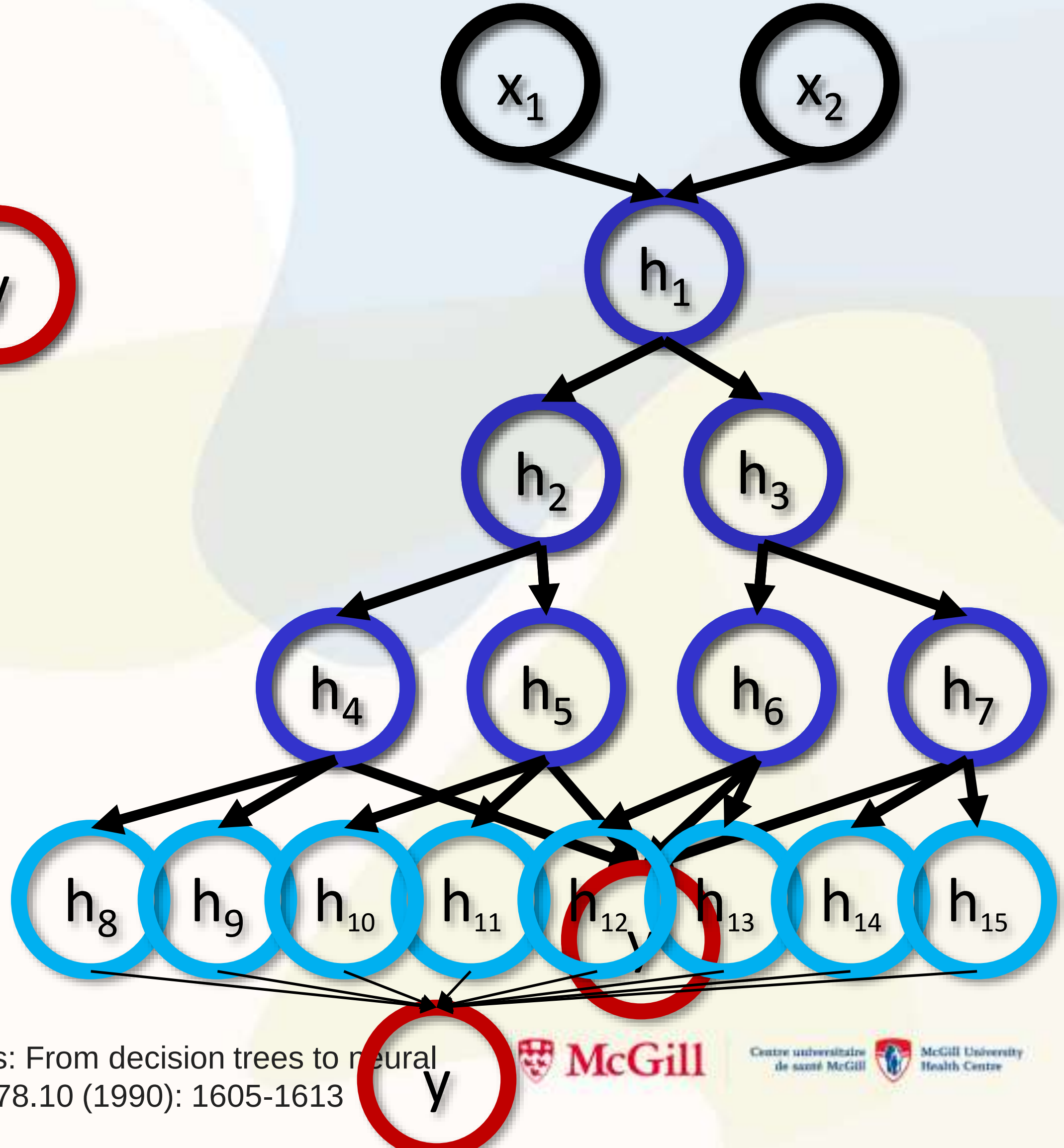




Sethi, Ishwar Krishnan. "Entropy nets: From decision trees to neural networks." *Proceedings of the IEEE* 78.10 (1990): 1605-1613

Neural Networks

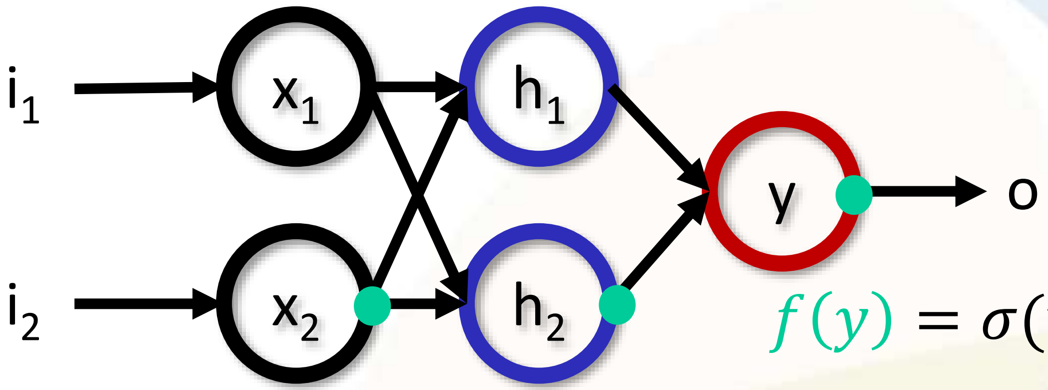




Sethi, Ishwar Krishnan. "Entropy nets: From decision trees to neural networks." *Proceedings of the IEEE* 78.10 (1990): 1605-1613

Neural Networks

Neural Networks



$$f(x_2) = \sigma(i_2 w_{x_2,i_2} + b_{x_2})$$

$$f(h_2) = \sigma(w_{h_2,x_1} f(x_1) + w_{h_2,x_2} f(x_2) + b_{h_2})$$

$$= \sigma(w_{h_2,x_1}\sigma(i_1 w_{x_1,i_1} + b_{x_1}) + w_{h_2,x_2}\sigma(i_2 w_{x_2,i_2} + b_{x_2}) + b_{h_2})$$

$$f(y) = \sigma(w_{y,h_1} f(h_1) + w_{y,h_2} f(h_2) + b_y)$$

$$= \sigma(w_{y,h_1}\sigma(w_{h_1,x_1}\sigma(i_1 w_{x_1} + b_{x_1}) + w_{1,x_2}\sigma(i_2 w_{x_2,i_2} + b_{x_2}) + b_{h_1}) + w_{y,h_2}\sigma(w_{h_2,x_1}\sigma(i_1 w_{x_1,i_1} + b_{x_1}) + w_{h_2,x_2}\sigma(i_2 w_{x_2,i_2} + b_{x_2}) + b_{h_2}) + b_y)$$

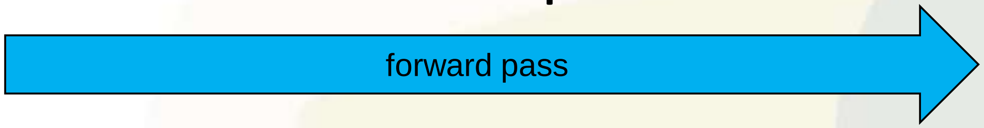
17 parameters $\theta = \{\boldsymbol{w}, \boldsymbol{b}\}$

Backpropagation

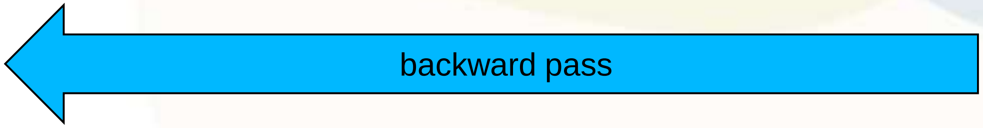
1. Random θ initialization

Iterate:

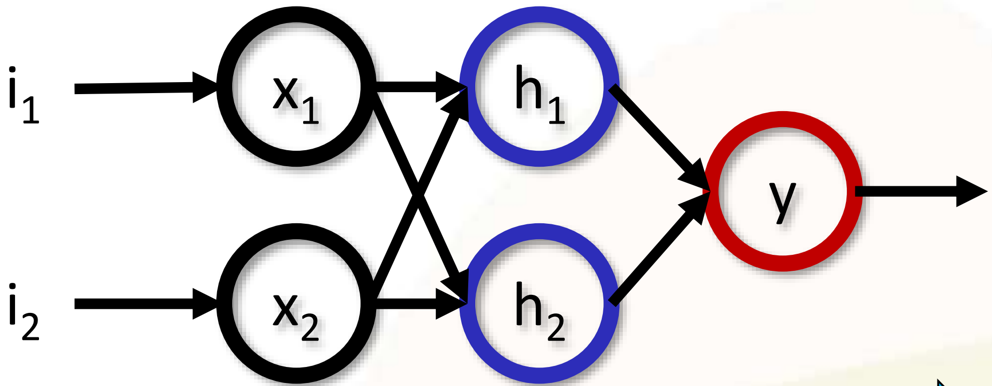
1. Forward - compute loss



2. Backward - update parameters



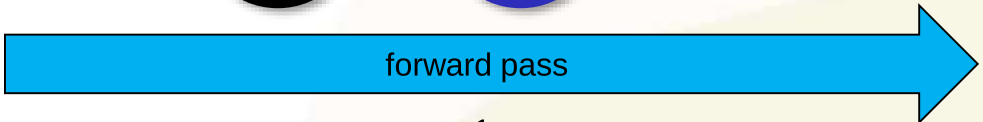
Backpropagation



$\hat{y} \approx P(o)$

XOR

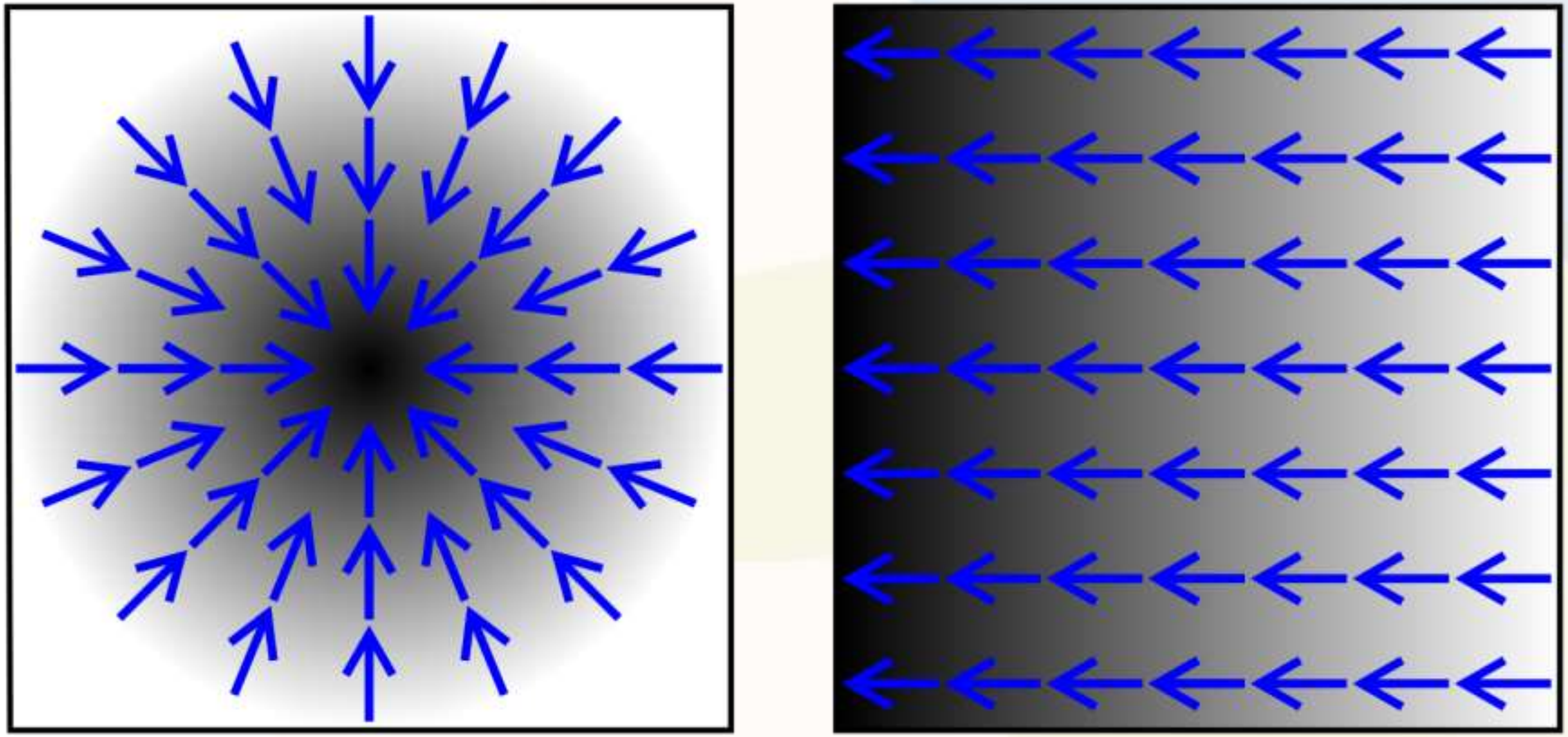
i_1	i_2	o
0	0	0
0	1	1
1	0	1
1	1	0



$$L(o, \hat{y}) = \frac{1}{2} \sum (o - \hat{y})^2$$

$$\nabla_\theta L(o, \hat{y}) = \left[\frac{\partial L}{\partial w_{x_1, i_1}}, \frac{\partial L}{\partial b_{x_1}}, \frac{\partial L}{\partial w_{x_2, i_2}}, \frac{\partial L}{\partial b_{x_2}}, \dots, \frac{\partial L}{\partial w_{y, h_2}} \right]^T$$

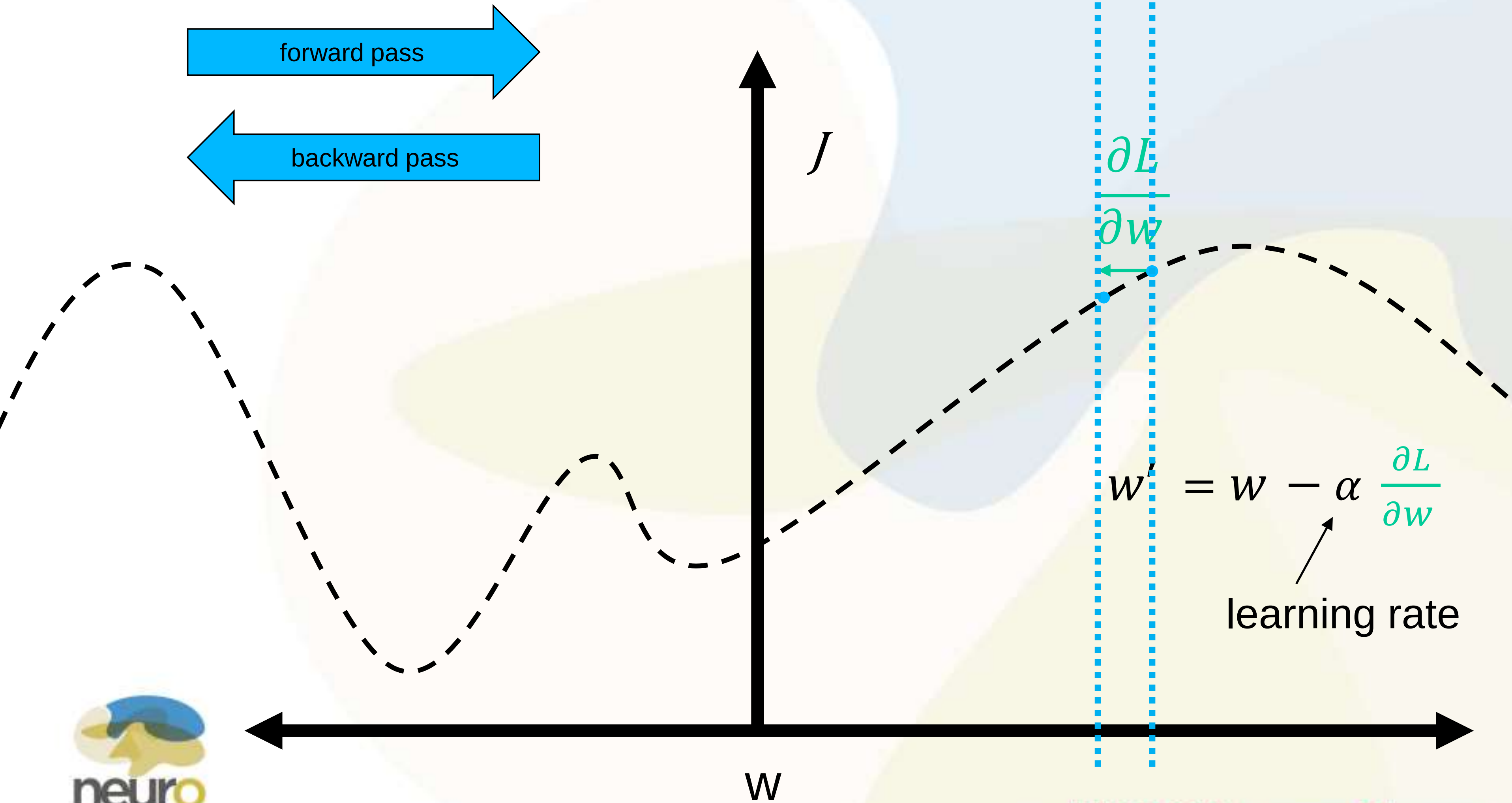
Backpropagation



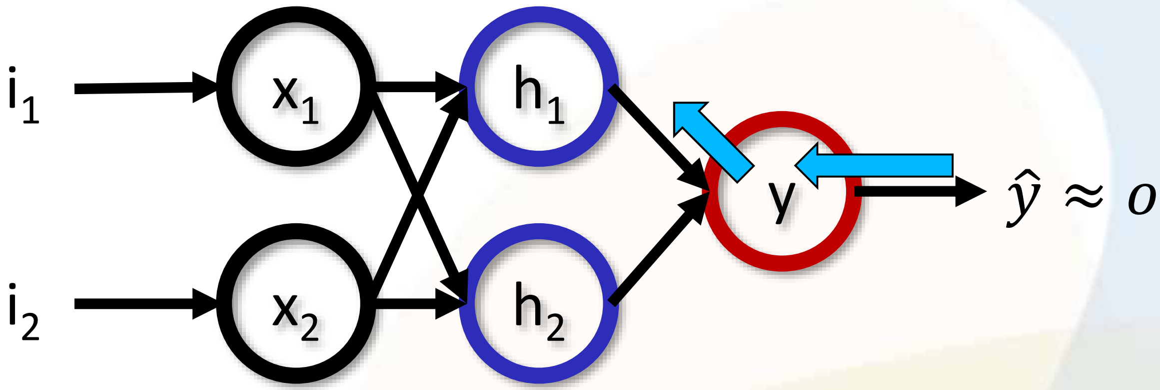
Gradients in blue

Backpropagation

Initialize w



Backpropagation

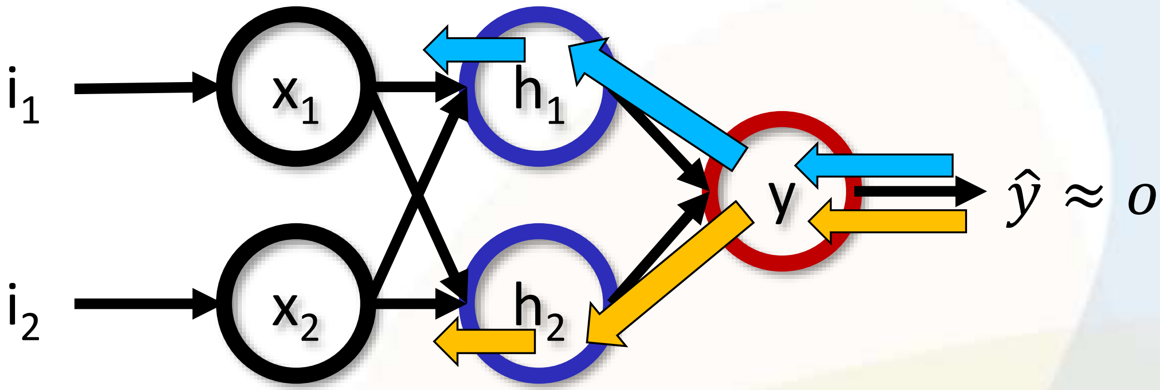


$$\frac{\partial L}{\partial w_{y,h_1}} = \frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial w_{y,h_1}}$$

$$\cdots$$

$$= \sum -\sigma(\hat{y})\big(1 - \sigma(\hat{y})\big)\, f(h_1)$$

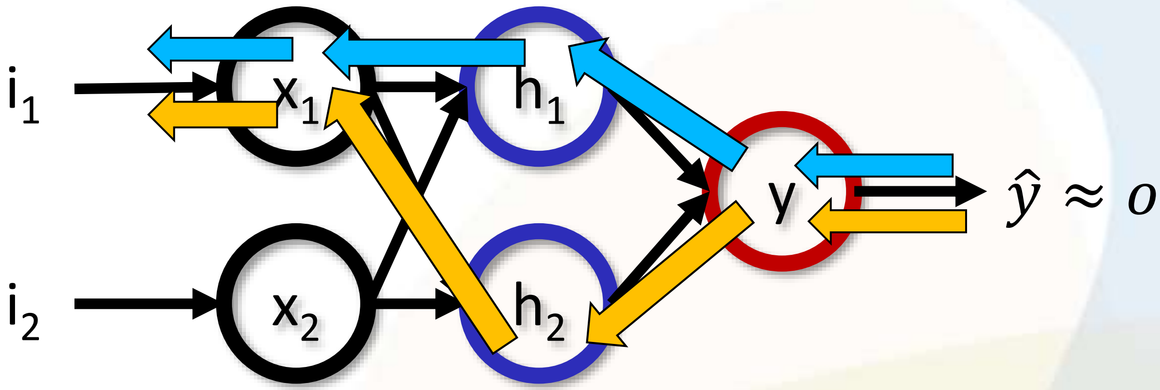
Backpropagation



$$\frac{\partial L}{\partial w_{h_1,x_1}} = \frac{\partial L}{\partial y} * \frac{\partial y}{\partial h_1} * \frac{\partial h_1}{\partial w_{h_1,x_1}}$$

$$\frac{\partial L}{\partial w_{h_2,x_2}} = \frac{\partial L}{\partial y} * \frac{\partial y}{\partial h_2} * \frac{\partial h_2}{\partial w_{h_2,x_2}}$$

Backpropagation



$$\frac{\partial L}{\partial w_{x_1,i_1}} = \frac{\partial L}{\partial y} * \frac{\partial y}{\partial h_1} * \frac{\partial h_1}{\partial x_1} * \frac{\partial x_1}{\partial w_{x_1,i_1}} + \frac{\partial L}{\partial y} * \frac{\partial y}{\partial h_2} * \frac{\partial h_2}{\partial x_1} * \frac{\partial x_1}{\partial w_{x_1,i_1}}$$

TensorFlow

PYTORCH

Optimizers

1. Gradient Descent $w' = w - \alpha \frac{\partial J}{\partial w}$
2. Stochastic Gradient Descent approx. $\frac{\partial J}{\partial w}$ in batches
3. Momentum $w' = w + v$ $v = \gamma v + \alpha \frac{\partial J}{\partial w}$
4. Adagrad/adadelta param-wise decaying learning rate
5. RMSprop avg. gradients
6. Adam RMSprop + momentum

Data Manifold

Data distribution:

- Class 1
- Class 2

X-Y grid:

- Param (θ) space

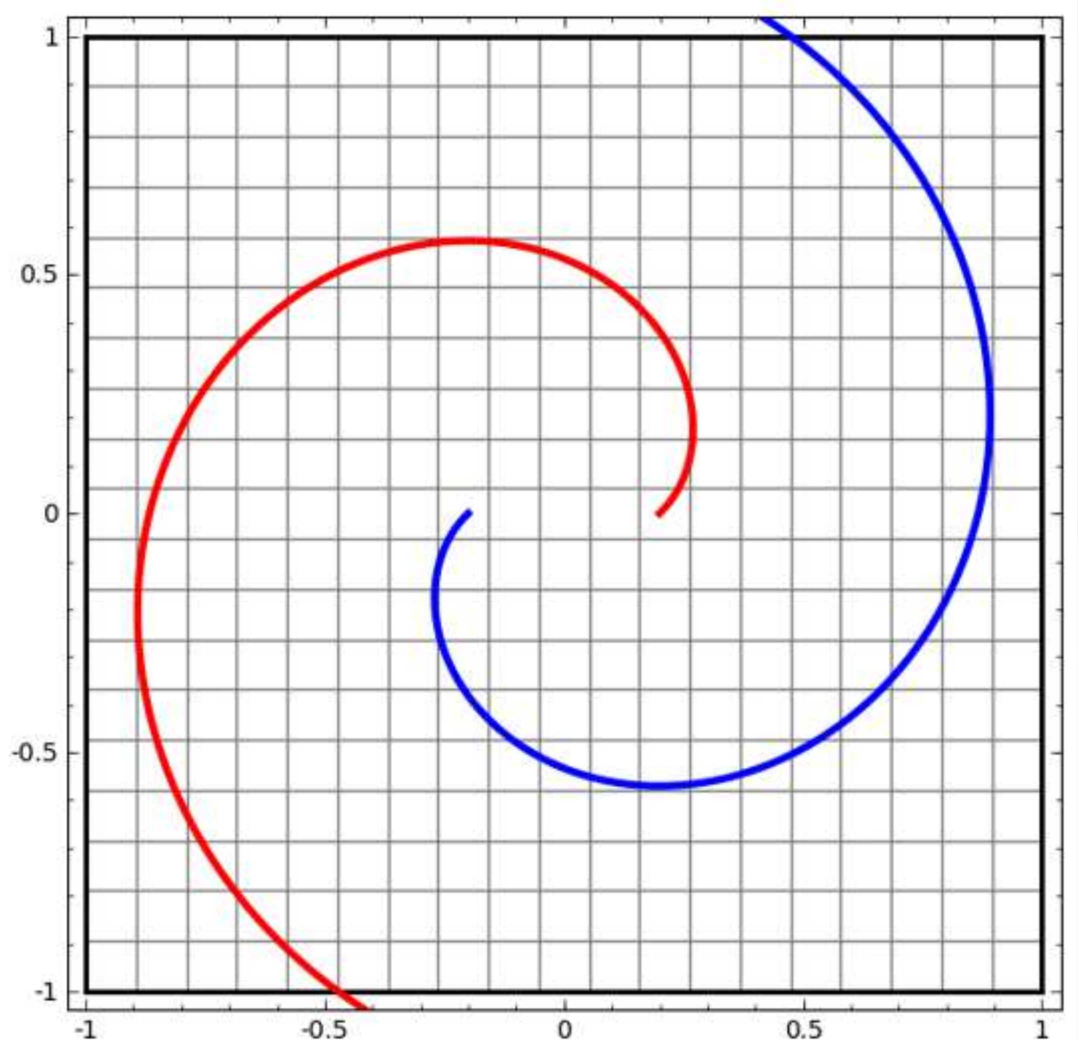


Image courtesy Chris Olah, 2014

Data Manifold

Data distribution:

- Class 1
- Class 2

X-Y grid:

- Param (θ) space

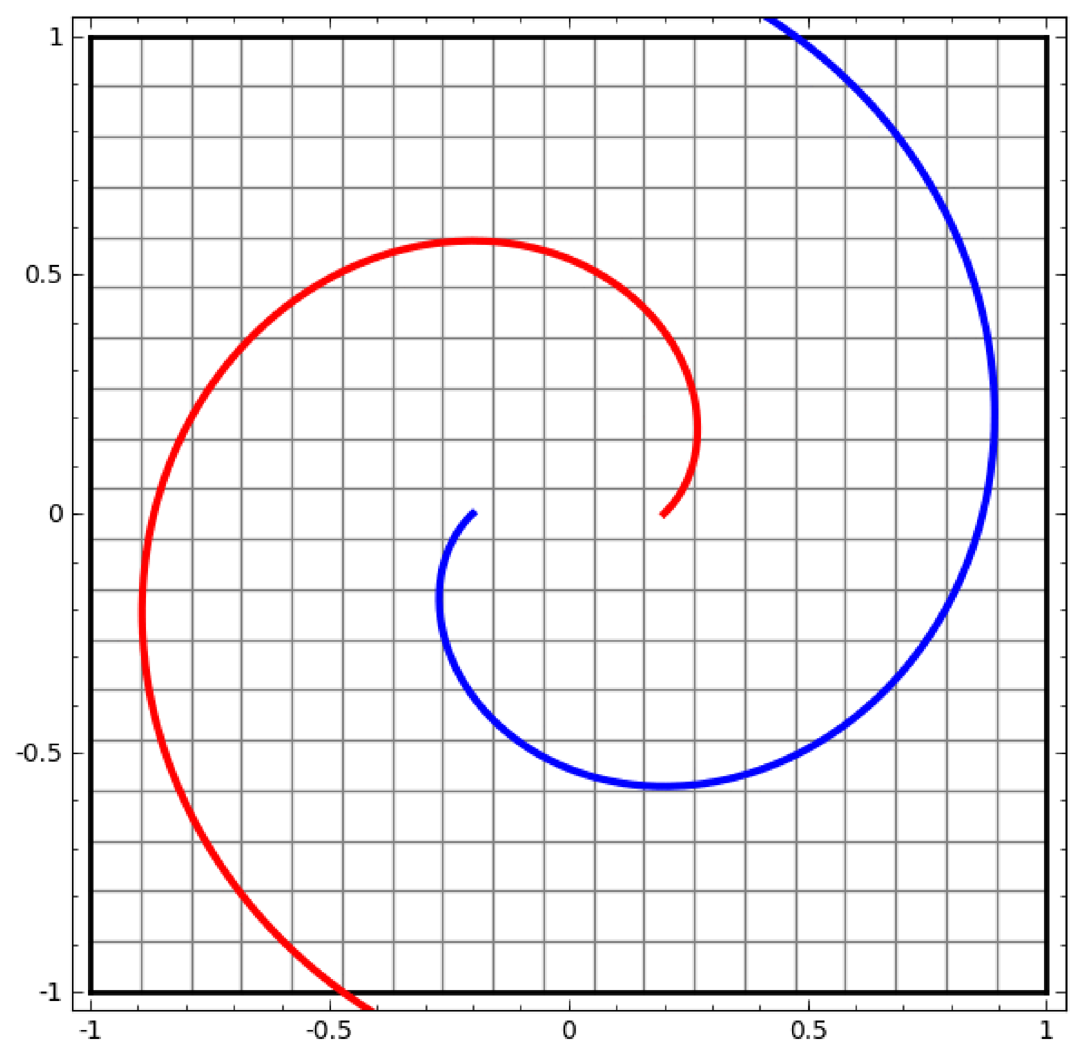


Image courtesy Chris Olah, 2014

Convolutional Neural Networks

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

Institut et hôpital neurologiques de Montréal
Montreal Neurological Institute and Hospital

McGill

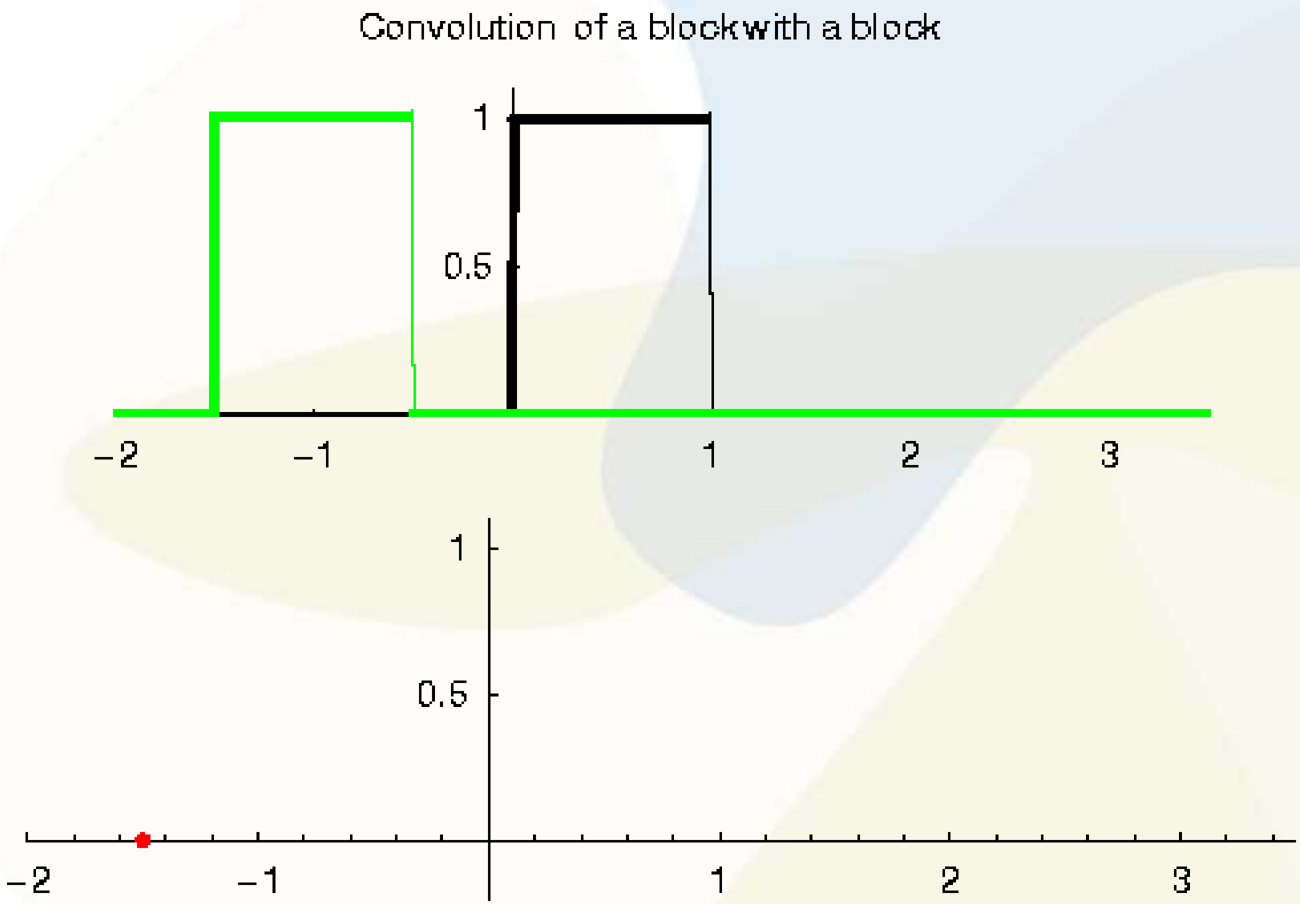
Centre universitaire de santé McGill
McGill University Health Centre

Convolutional Neural Networks

$g(t)$

$f(t)$

$f(t) * g(t)$

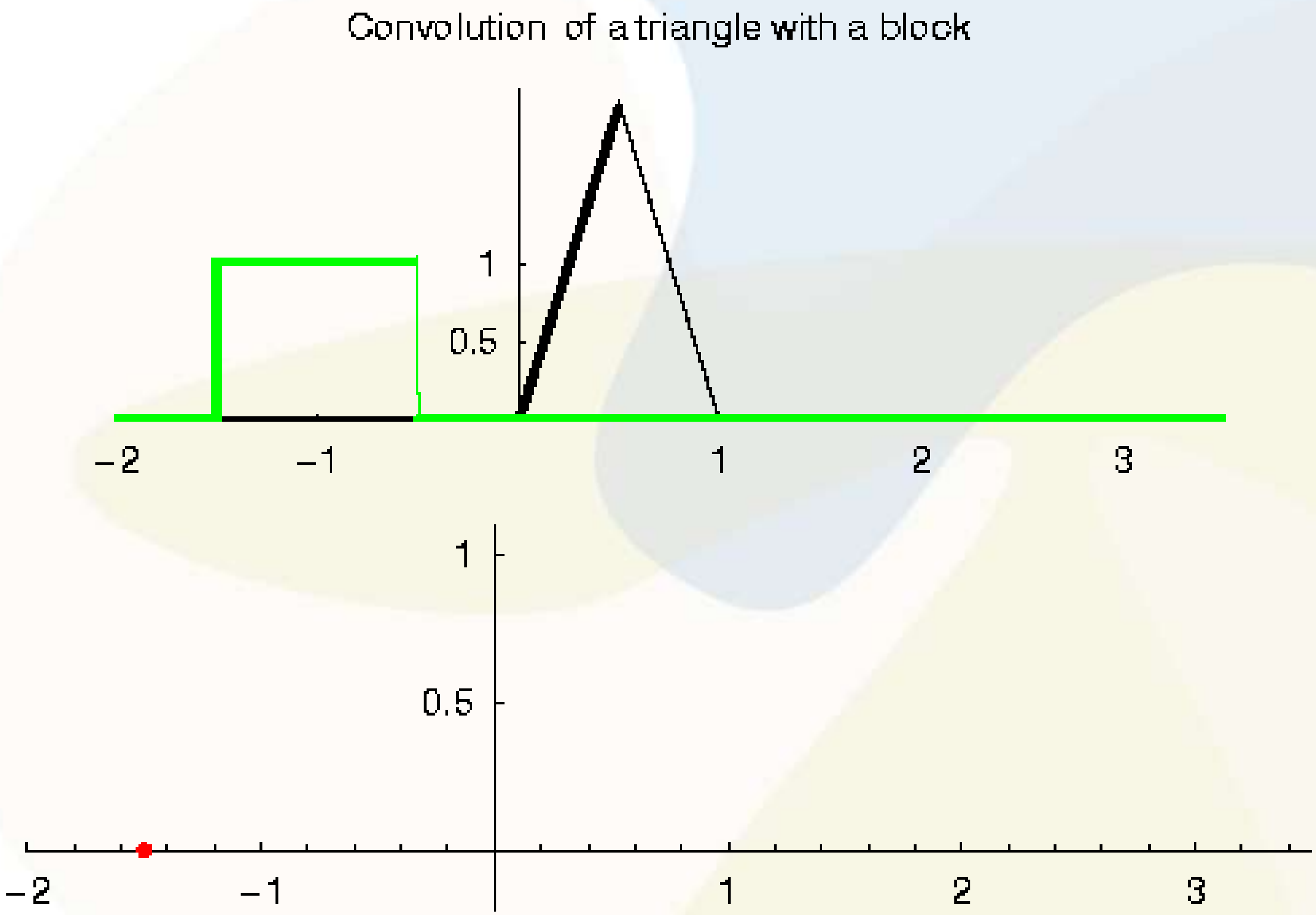


Bart M. ter Haar Romeny. University of Technology Eindhoven, 1998.
http://bmia.bmt.tue.nl/education/courses/fev/course/notebooks/Convolution.html

Convolutional Neural Networks

$g(t)$

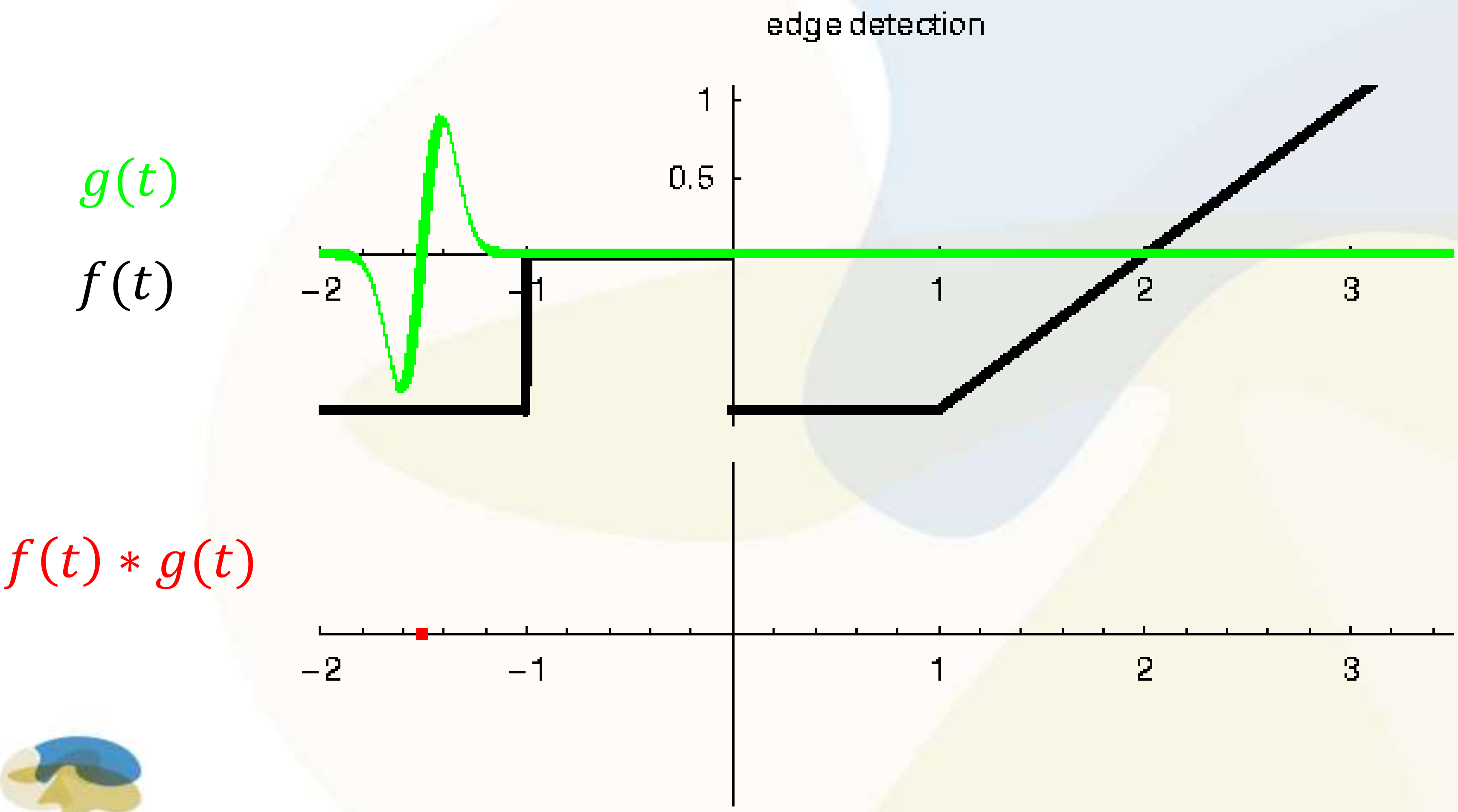
$f(t)$



$f(t) * g(t)$

Bart M. ter Haar Romeny. University of Technology Eindhoven, 1998.
http://bmia.bmt.tue.nl/education/courses/fev/course/notebooks/Convolution.html

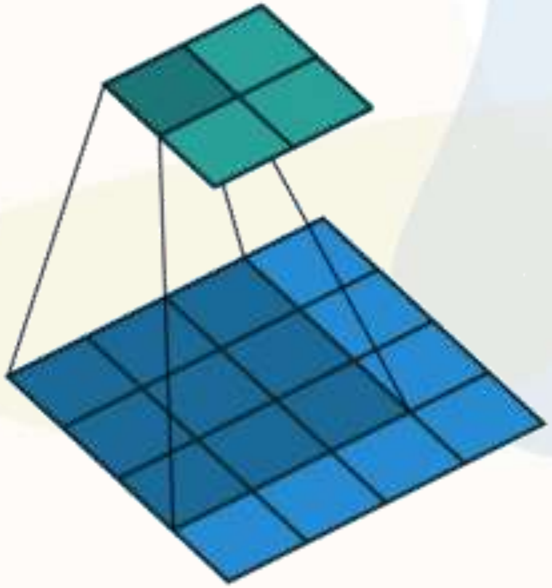
Convolutional Neural Networks



Bart M. ter Haar Romeny. University of Technology Eindhoven, 1998.
http://bmia.bmt.tue.nl/education/courses/fev/course/notebooks/Convolution.html

Convolutional Neural Networks

Output



Input

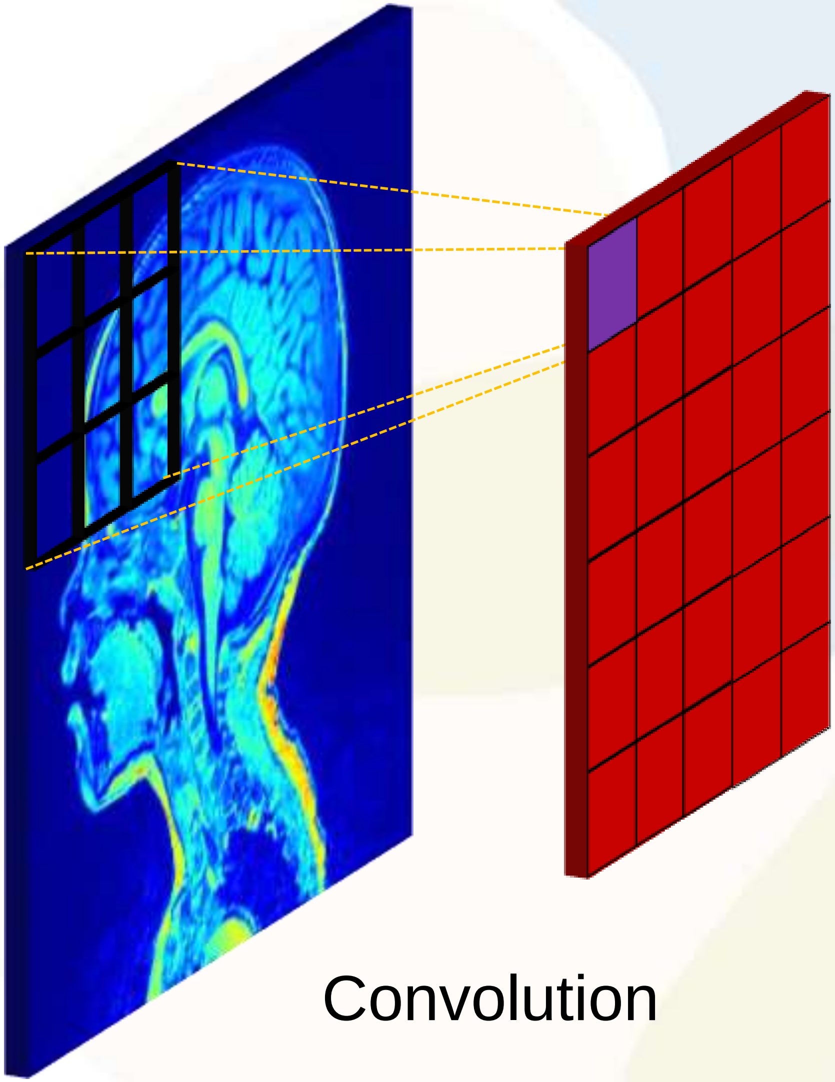
Dumoulin, Vincent, and Francesco Visin. "A guide to convolution arithmetic for deep learning." *arXiv preprint arXiv:1603.07285* (2016).

Convolutional Neural Networks

CNN/convnet neurons:

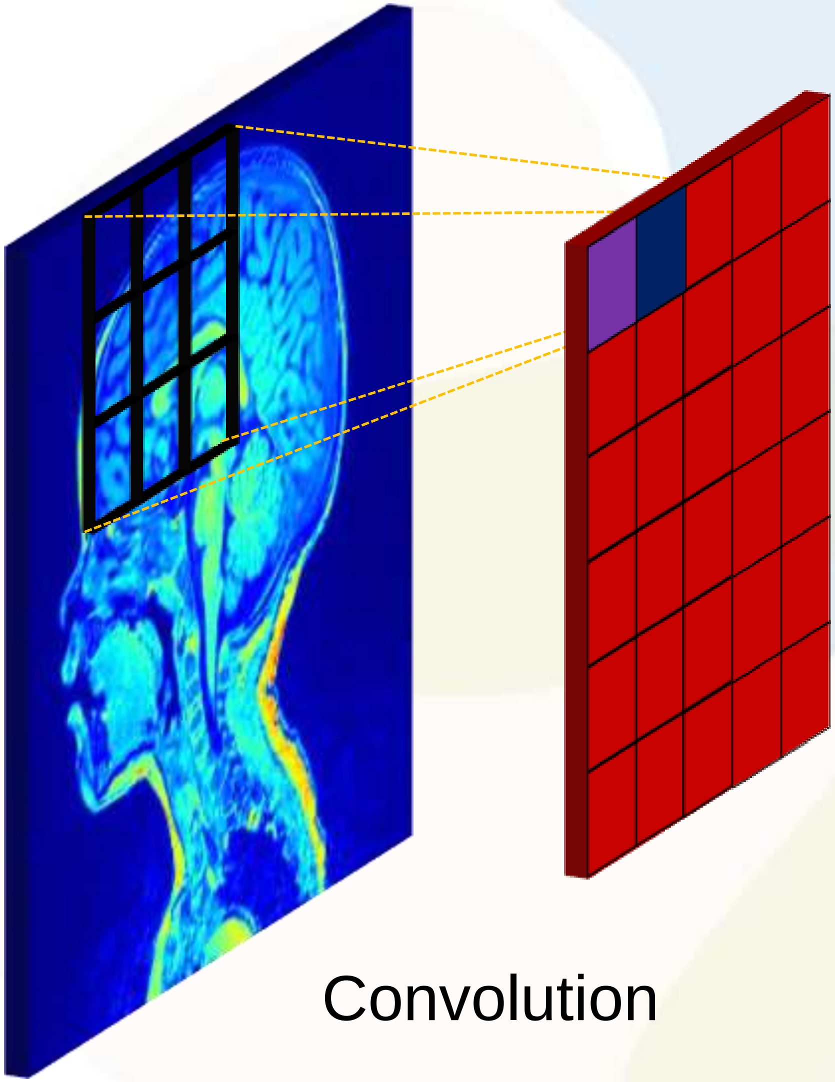
1. Have receptive fields
2. Share weights
    - Vastly reduces parameters
    - Translational equivariance

Convolutional Neural Networks



Convolution

Convolutional Neural Networks



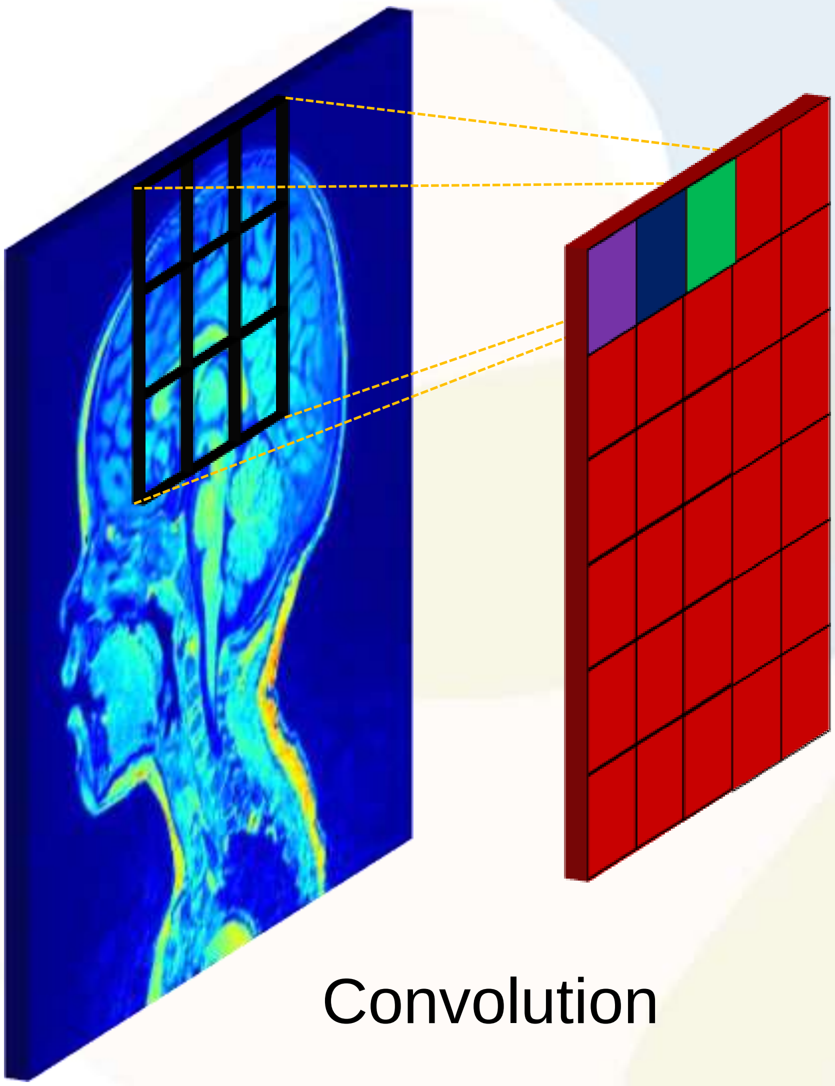
Convolution

neuro
Institut et hôpital neurologiques de Montréal
Montreal Neurological Institute and Hospital

McGill

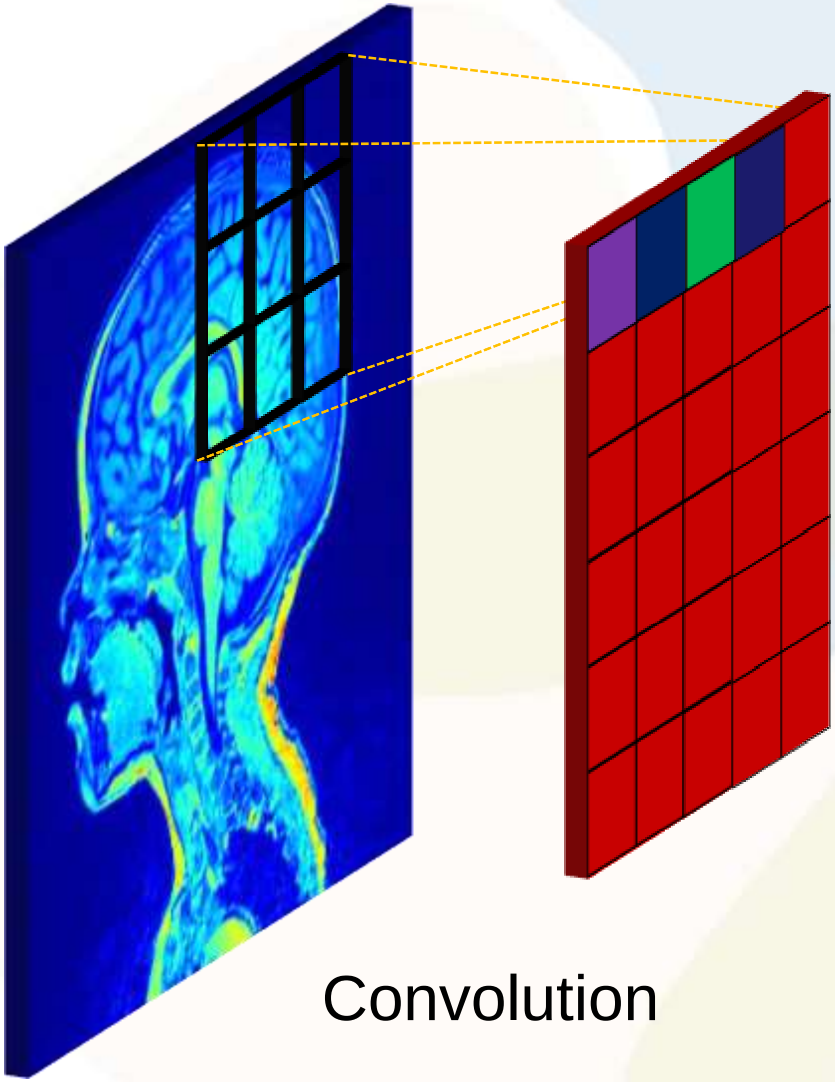
Centre universitaire de santé McGill
McGill University Health Centre

Convolutional Neural Networks



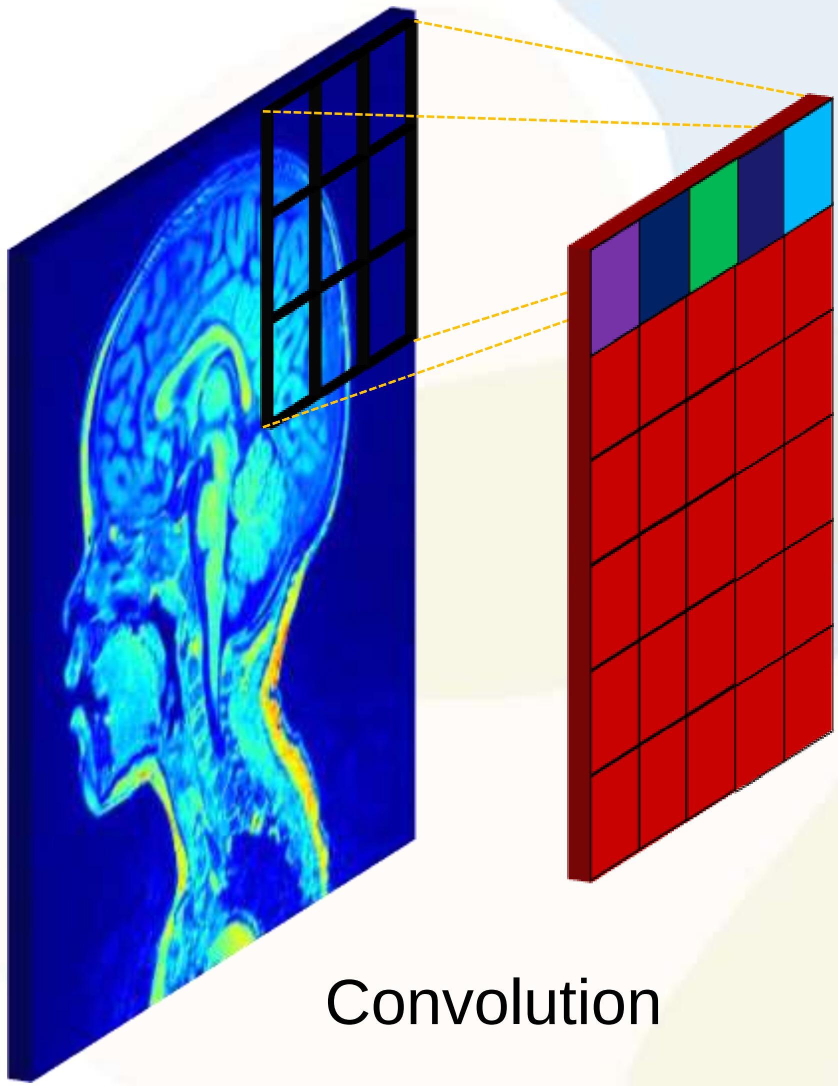
Convolution

Convolutional Neural Networks

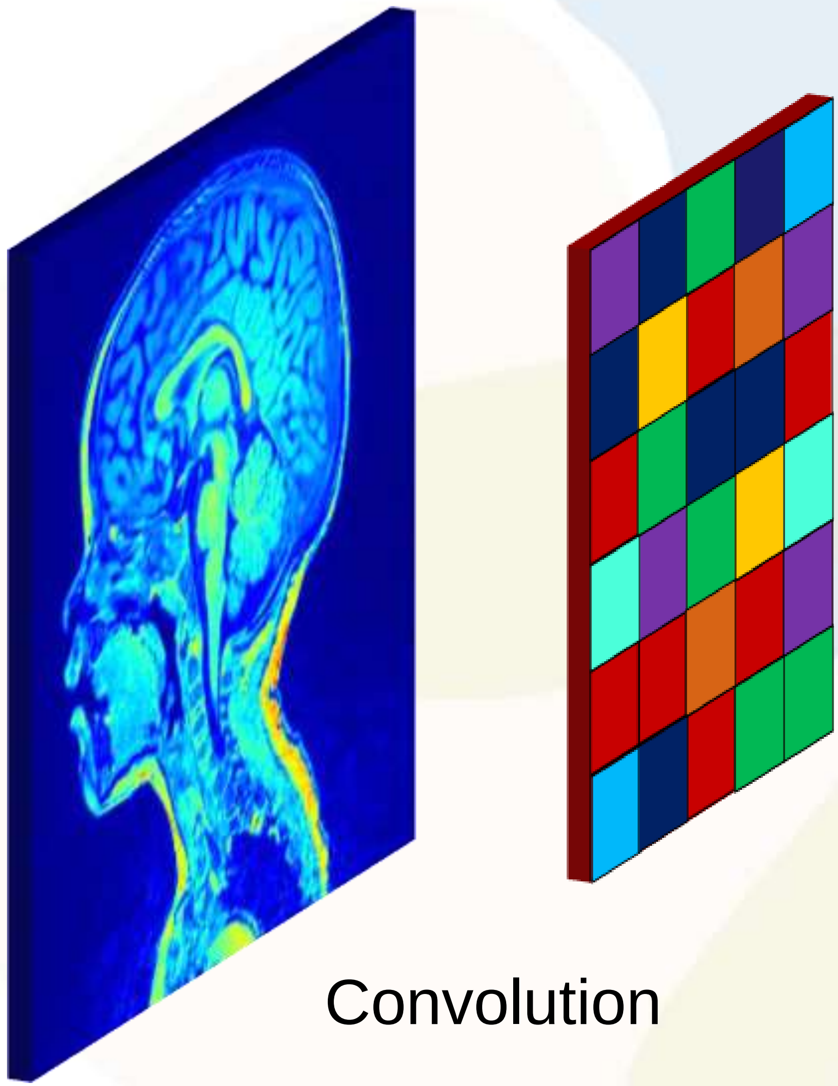


Convolution

Convolutional Neural Networks

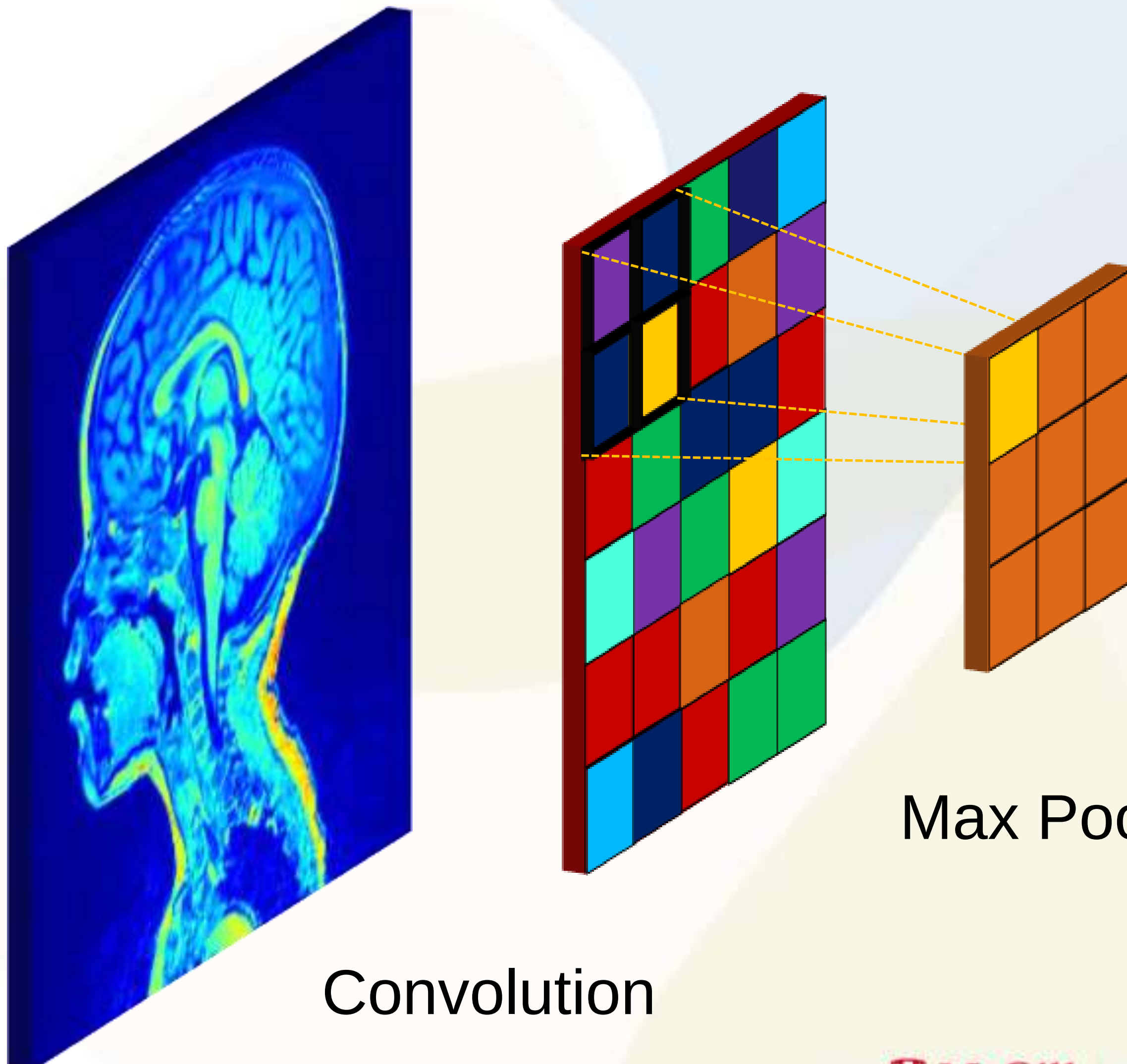


Convolutional Neural Networks

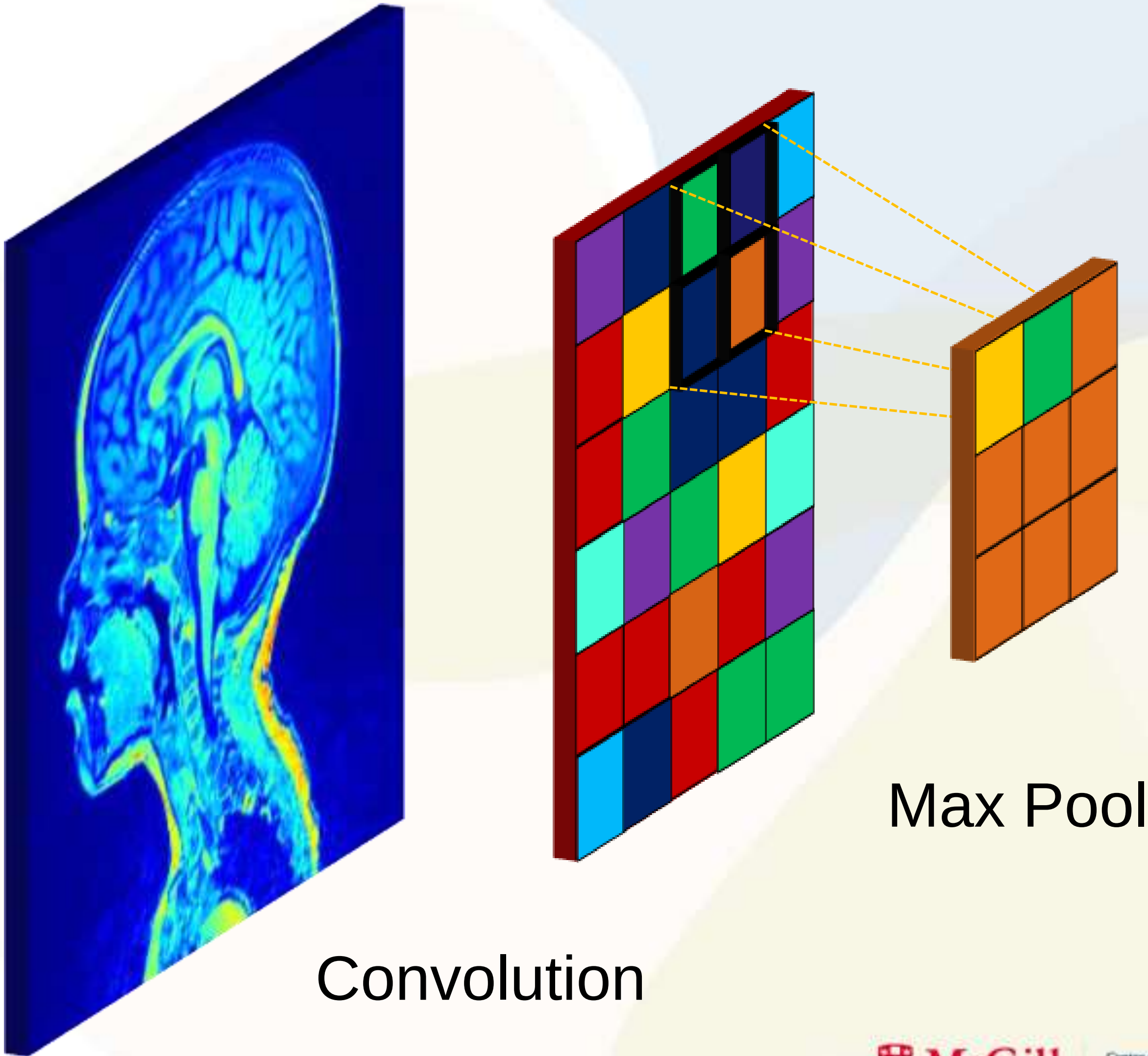


Convolution

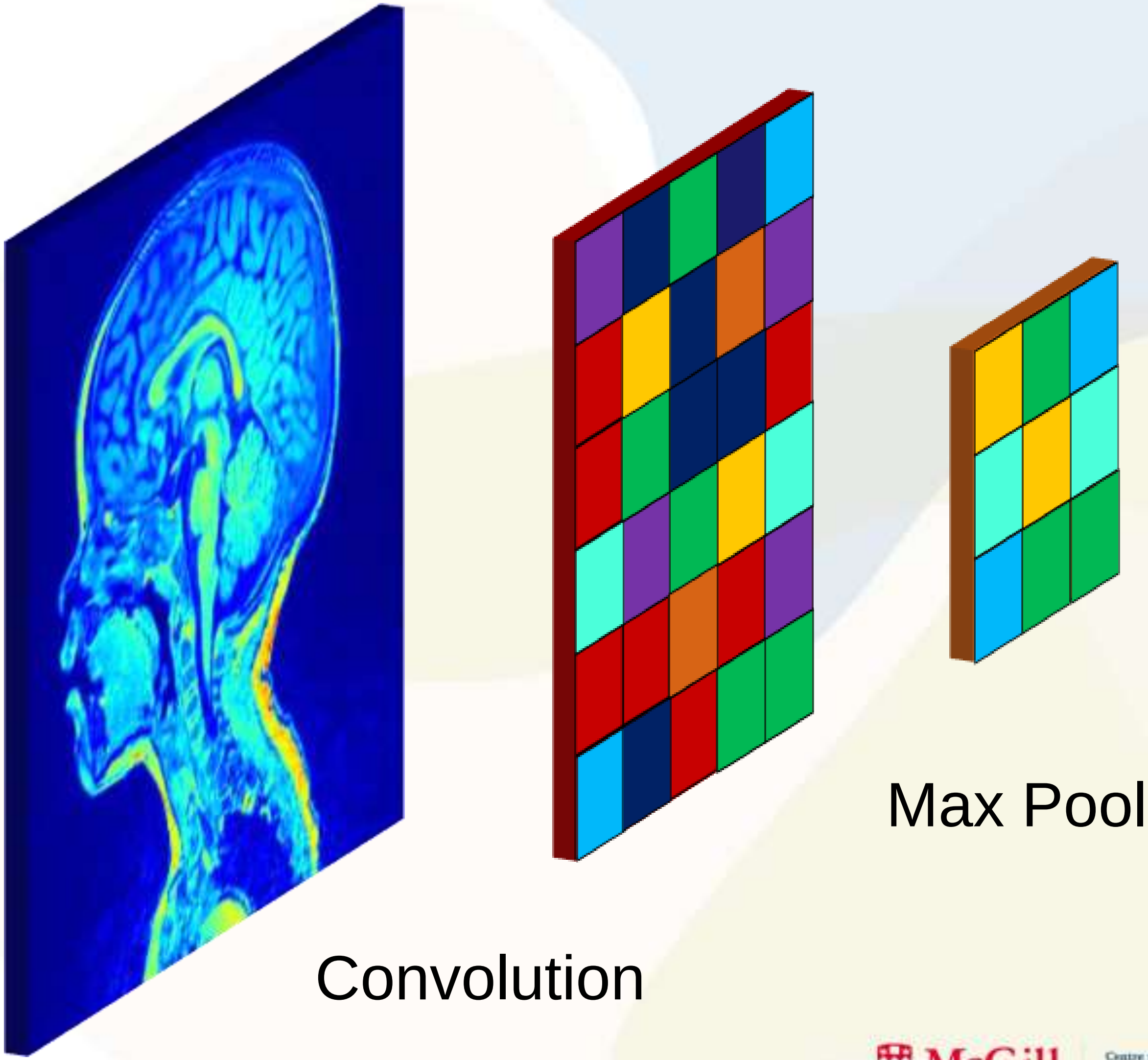
Convolutional Neural Networks



Convolutional Neural Networks

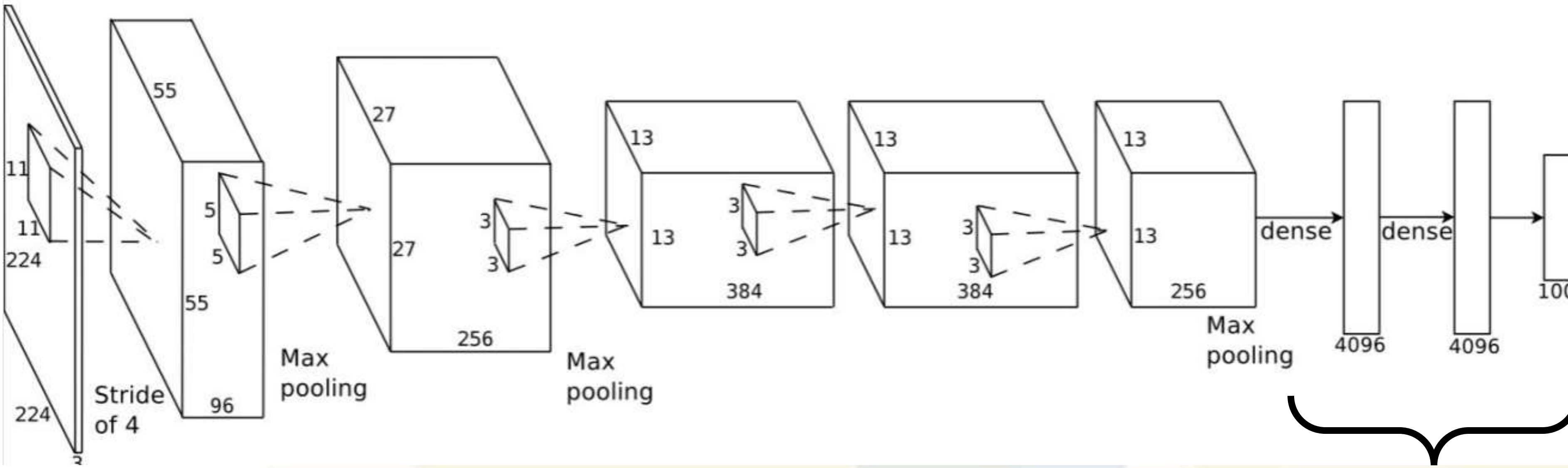


Convolutional Neural Networks



Institut et hôpital neurologiques de Montréal
Montreal Neurological Institute and Hospital

Convolutional Neural Networks



90% parameters

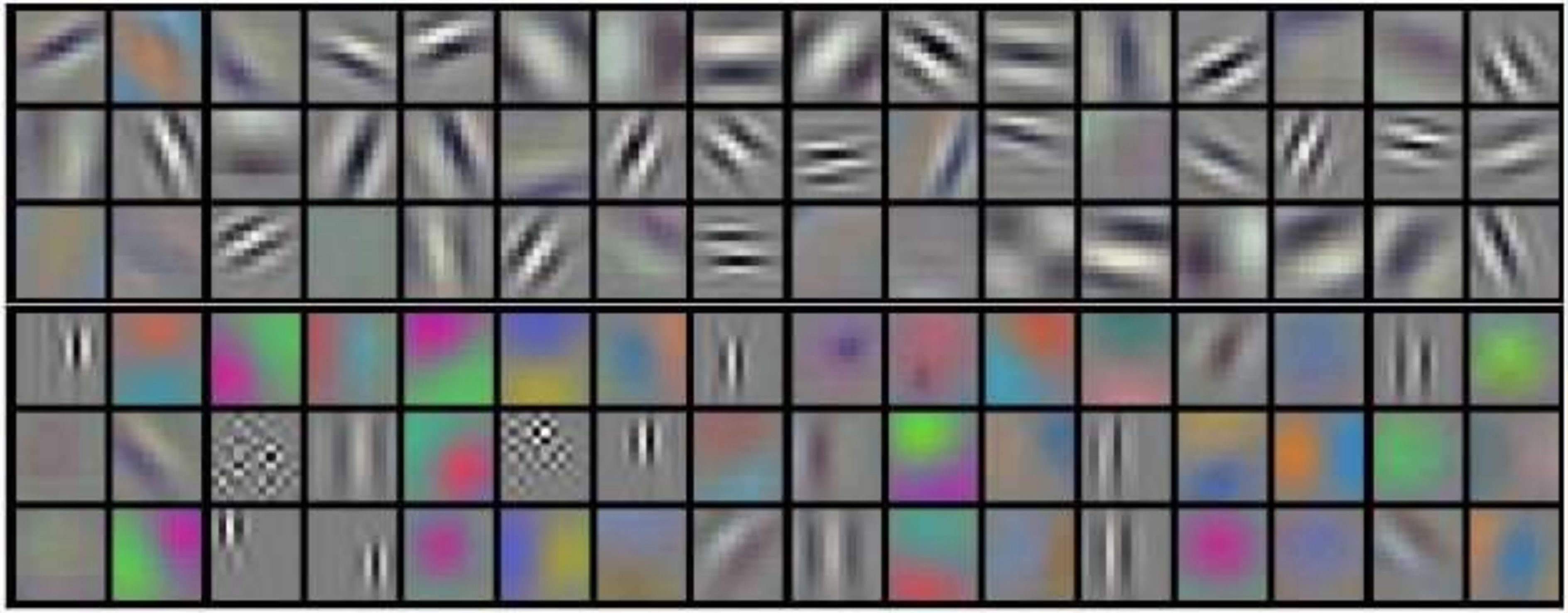
AlexNet trained using **Dropout**

neuro

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

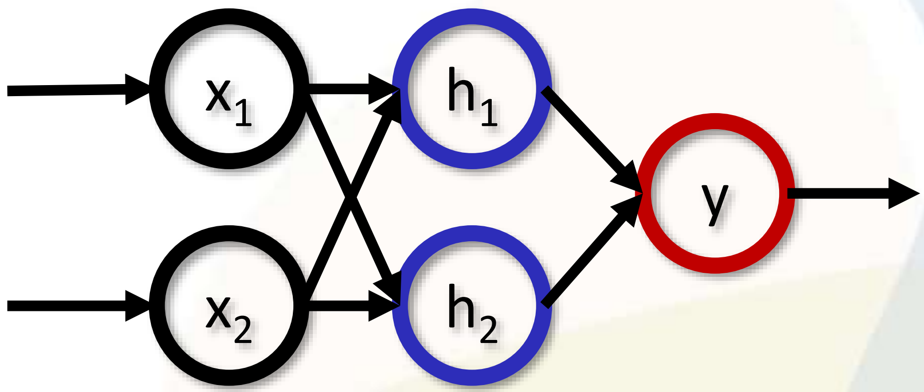
McGill

Convolutional Neural Networks



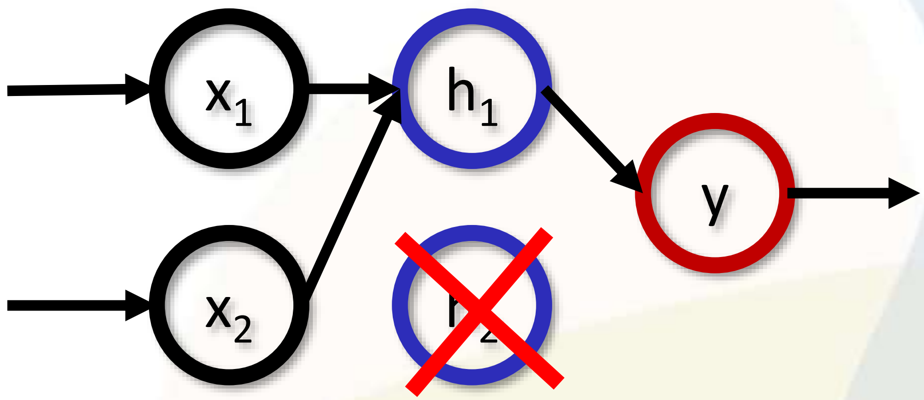
Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

Dropout



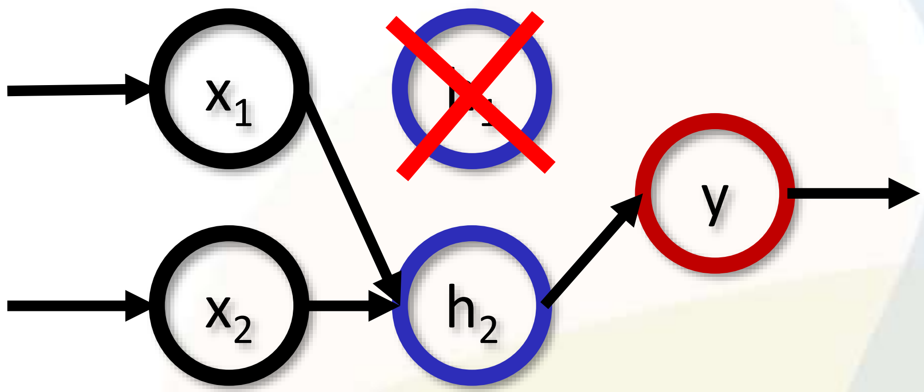
P(dropout) = 0.5

Dropout



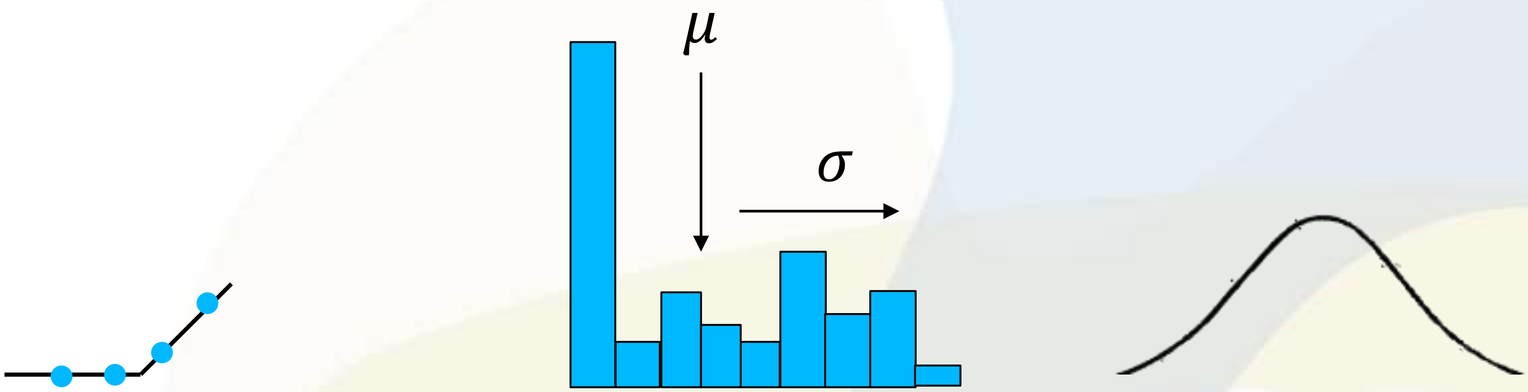
P(dropout) = 0.5

Dropout



P(dropout) = 0.5

Batch Normalization



- Subtract mean, divide by standard deviation

Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *arXiv preprint arXiv:1502.03167* (2015).

Batch Normalization

- Whitens activations
- Speeds training
- Injects noise

- Not good for small batch sizes

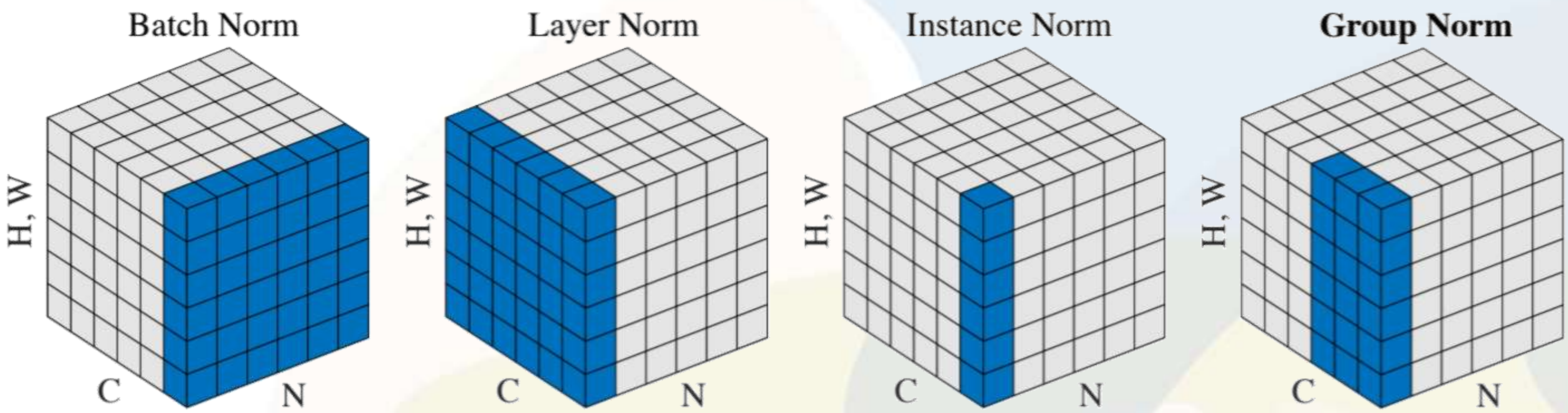
Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." *arXiv preprint arXiv:1502.03167* (2015).

neuro
Institut et hôpital neurologiques de Montréal
Montreal Neurological Institute and Hospital

McGill

Centre universitaire de santé McGill
McGill University Health Centre

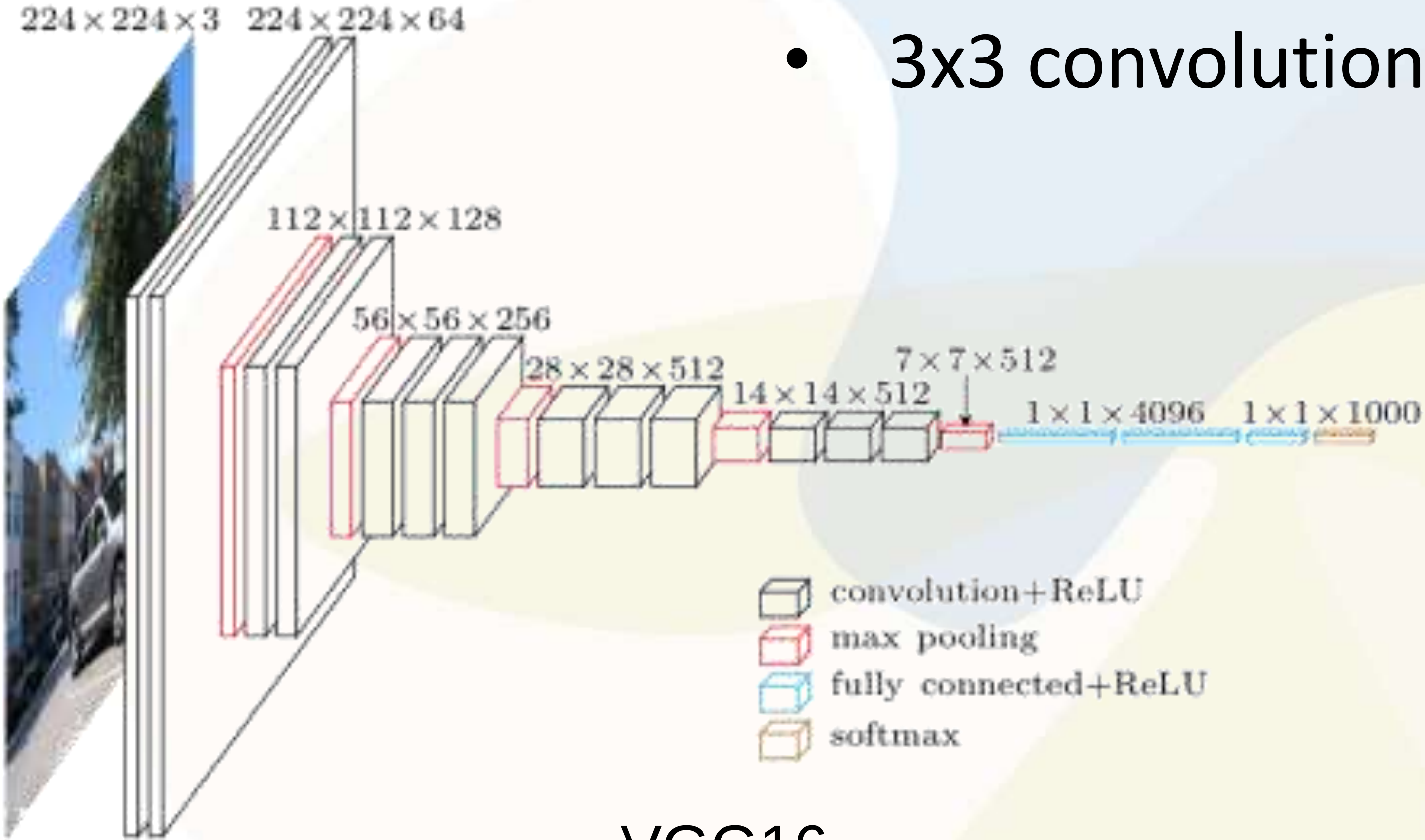
Group Normalization



- N : training examples
- C : channels
- H, W : spatial dimensions

Wu, Yuxin, and Kaiming He. "Group normalization." *arXiv preprint arXiv:1803.08494* (2018).

Convolutional Neural Networks



- 3x3 convolutions

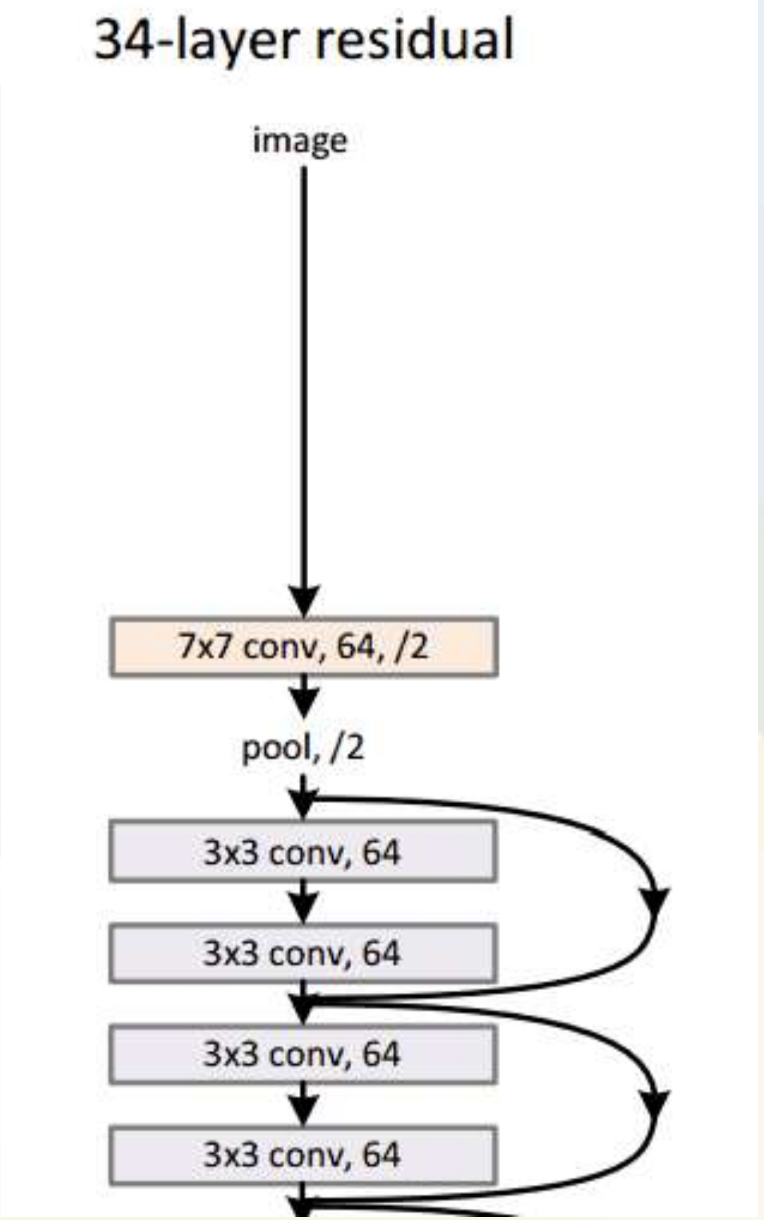
VGG16

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

ResNet

152 convolutional layers

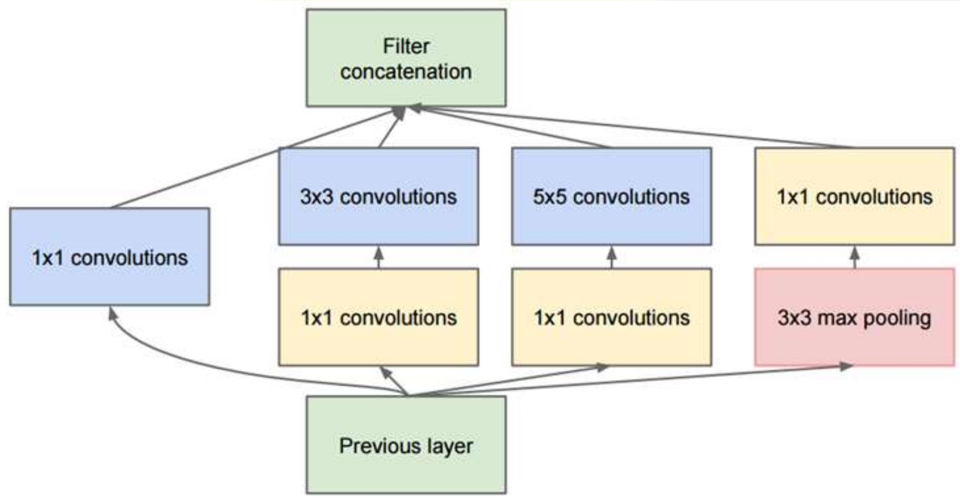
Skip (residual) connections



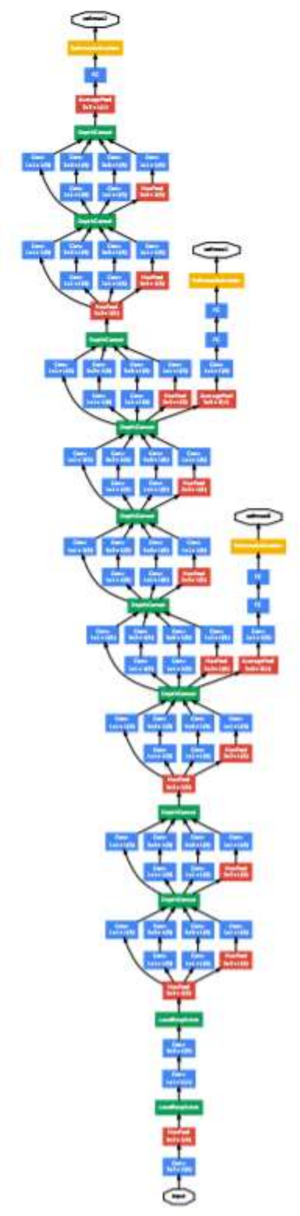
He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

GoogLeNet

1. Deep Supervision helps training
2. 1x1 convolutions can replace fully-connected layers

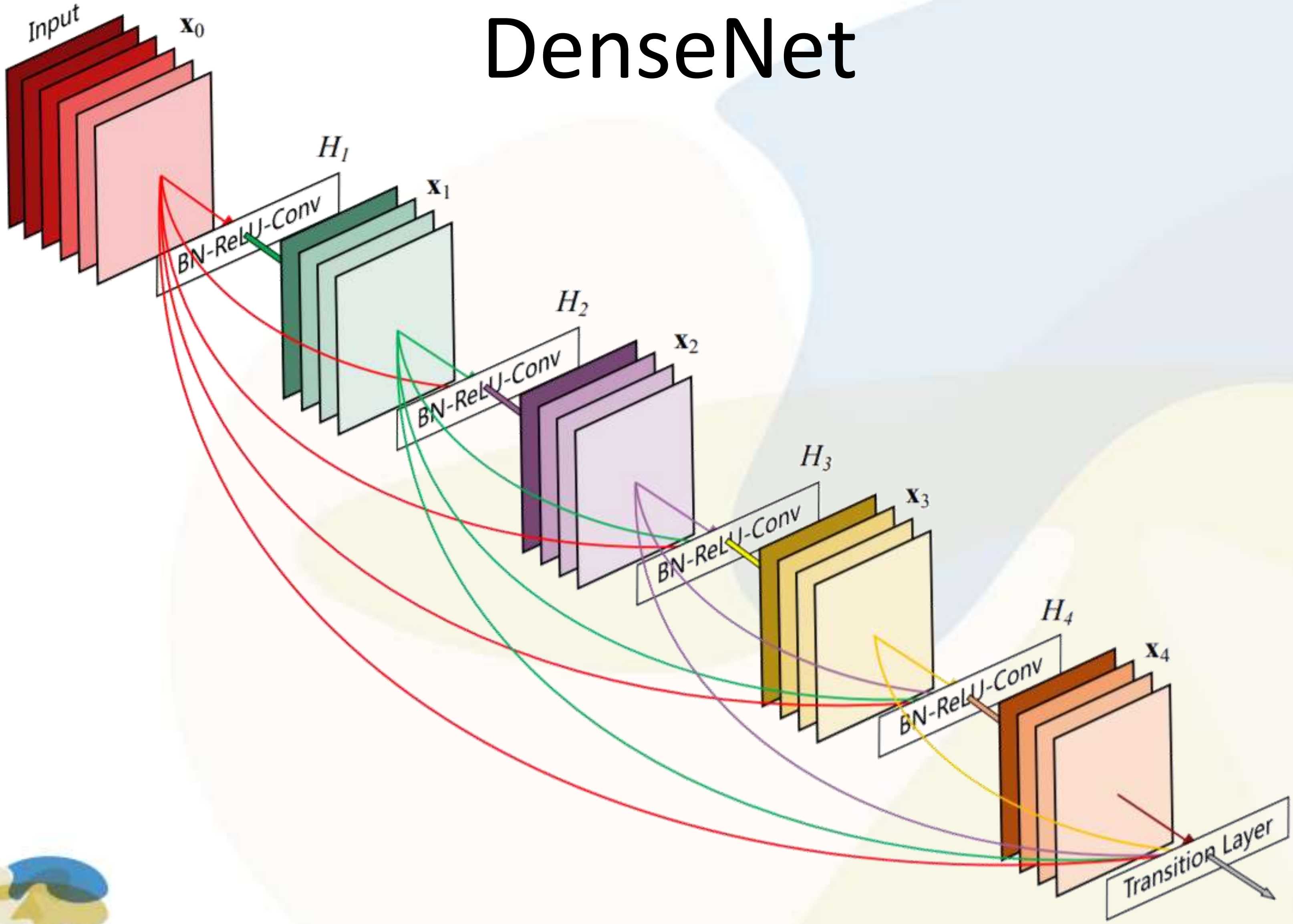


Full Inception module



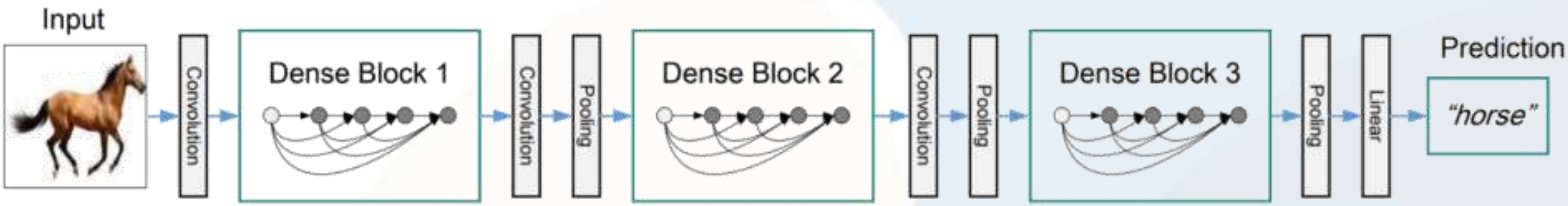
Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

DenseNet



Huang, Gao, et al. "Densely connected convolutional networks." *CVPR*. Vol. 1. No. 2. 2017.

DenseNet



- Densely-connected blocks & transition layers
- Far more parameter-efficient
- Doesn't need fancy optimizers

Huang, Gao, et al. "Densely connected convolutional networks." *CVPR*. Vol. 1. No. 2. 2017.

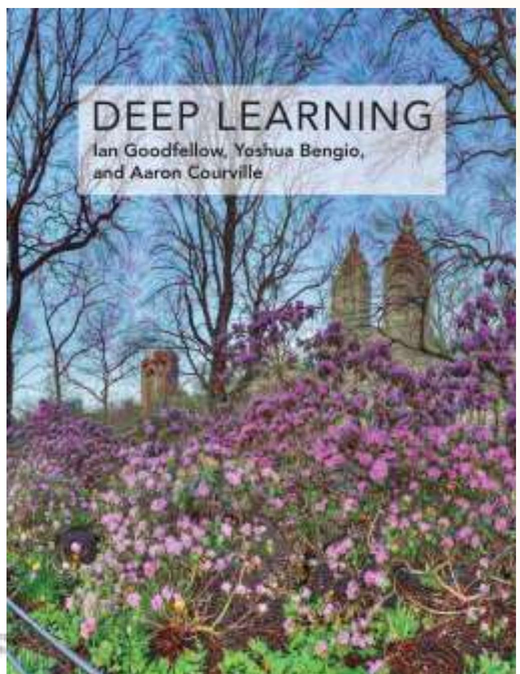
Challenges

1. Data quantity
2. Data size
3. Data quality
4. Data variability
5. Unexpected pathology

Start here

Beyond Linear Decoding
Introduction to Deep Learning

bit.ly/ohbmdl



Keras

keras.io

deeplearningbook.org

Thanks!

neuro
Institut et hôpital neurologiques de Montréal
Montreal Neurological Institute and Hospital

McGill

Centre universitaire de santé McGill
McGill University Health Centre