Multiple Shades of Dropout for Discriminative and Generative Deep Neural Networks

Boqing Gong

Tencent AI Lab, Seattle

BoqingGo@outlook.com

Research overview

Computer vision

Machine learning

2D/3D object recognition
Video summarization
Human activity recognition
Image tagging
Semantic segmentation
Face detection
Visual question answering

Domain adaptation & kernel methods
Multi-task & Transfer learning
zero-shot learning
Deep learning
Probabilistic models

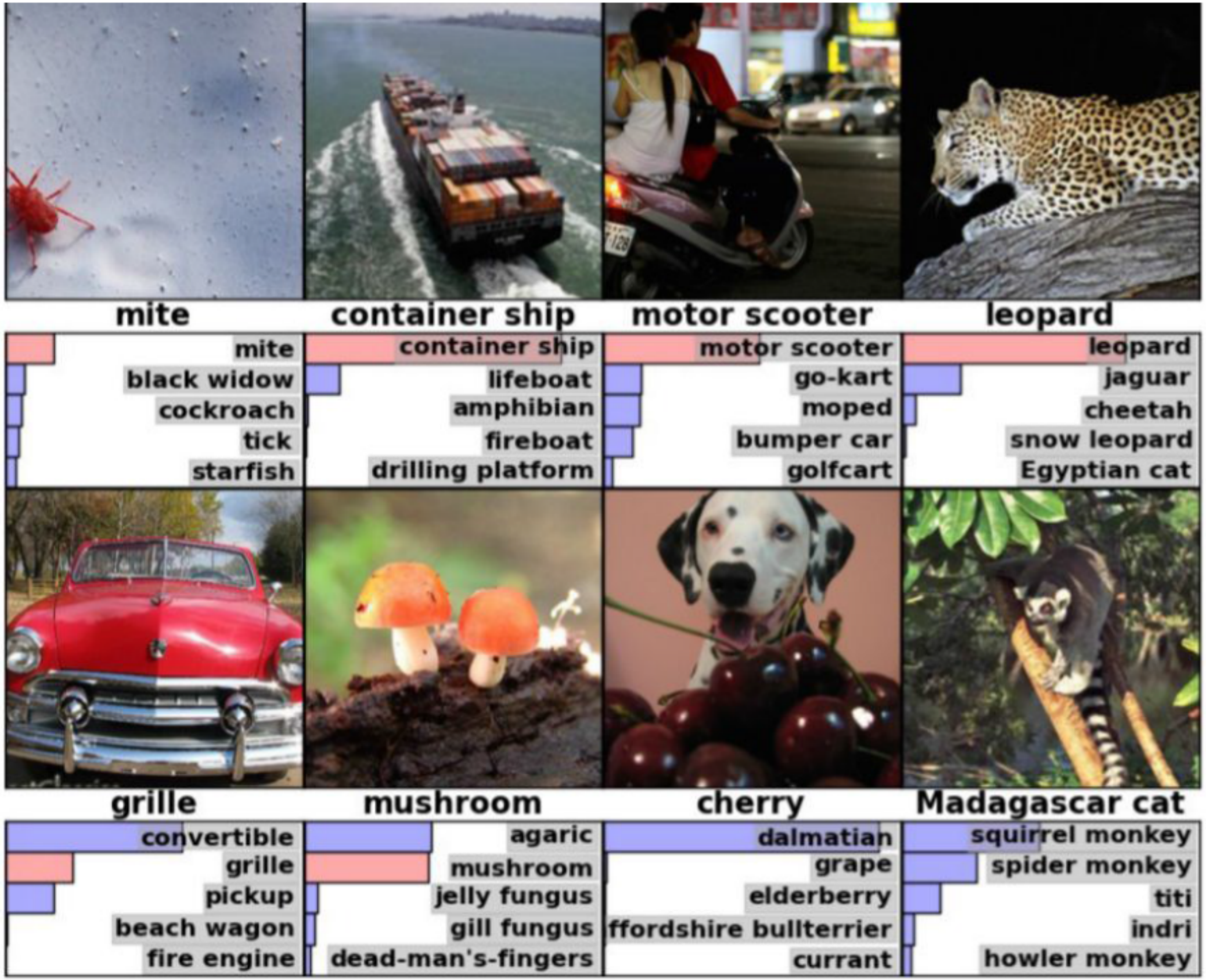
Object Recognition

Correct label

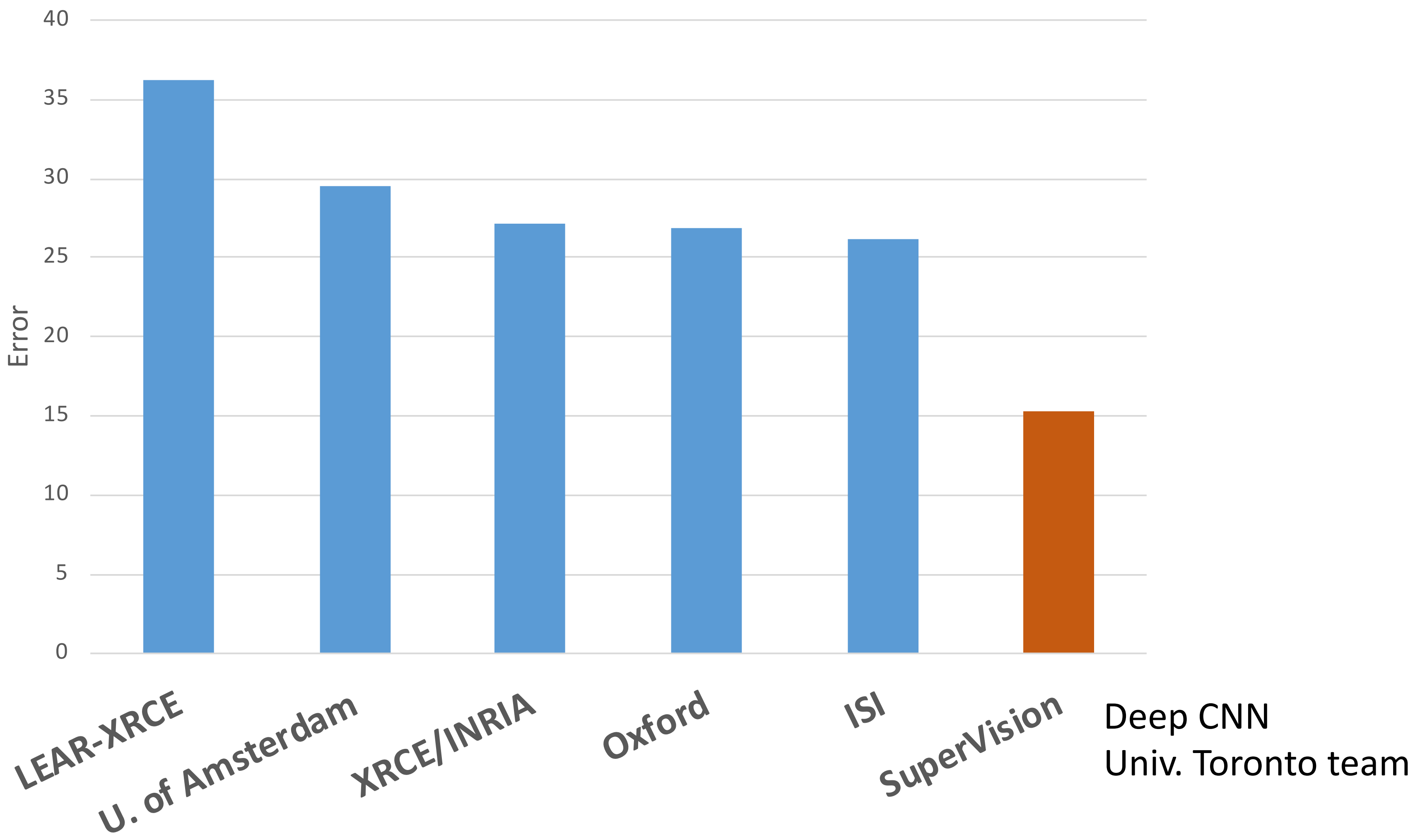
correct predictions

Correct label

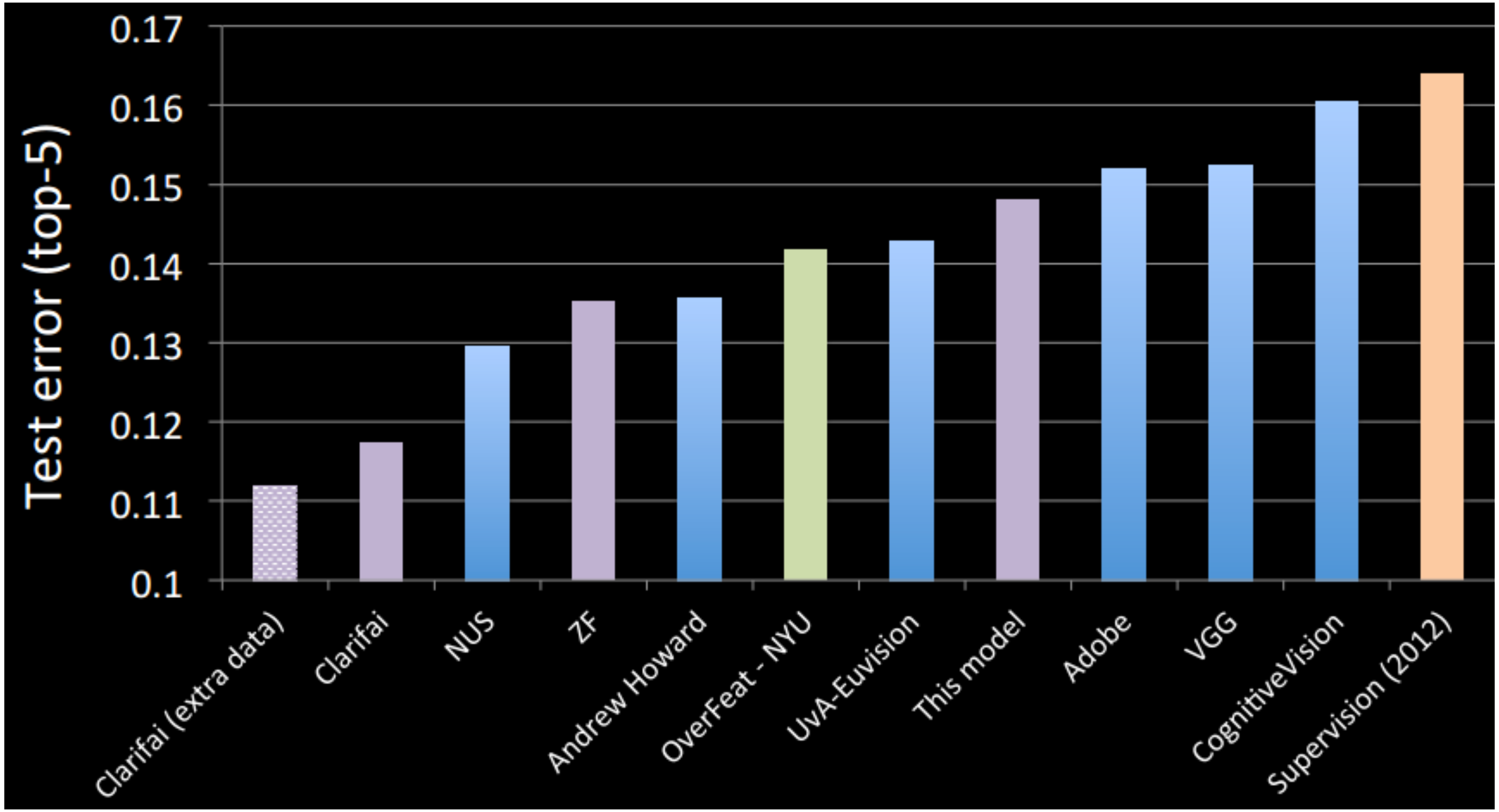
incorrect? predictions



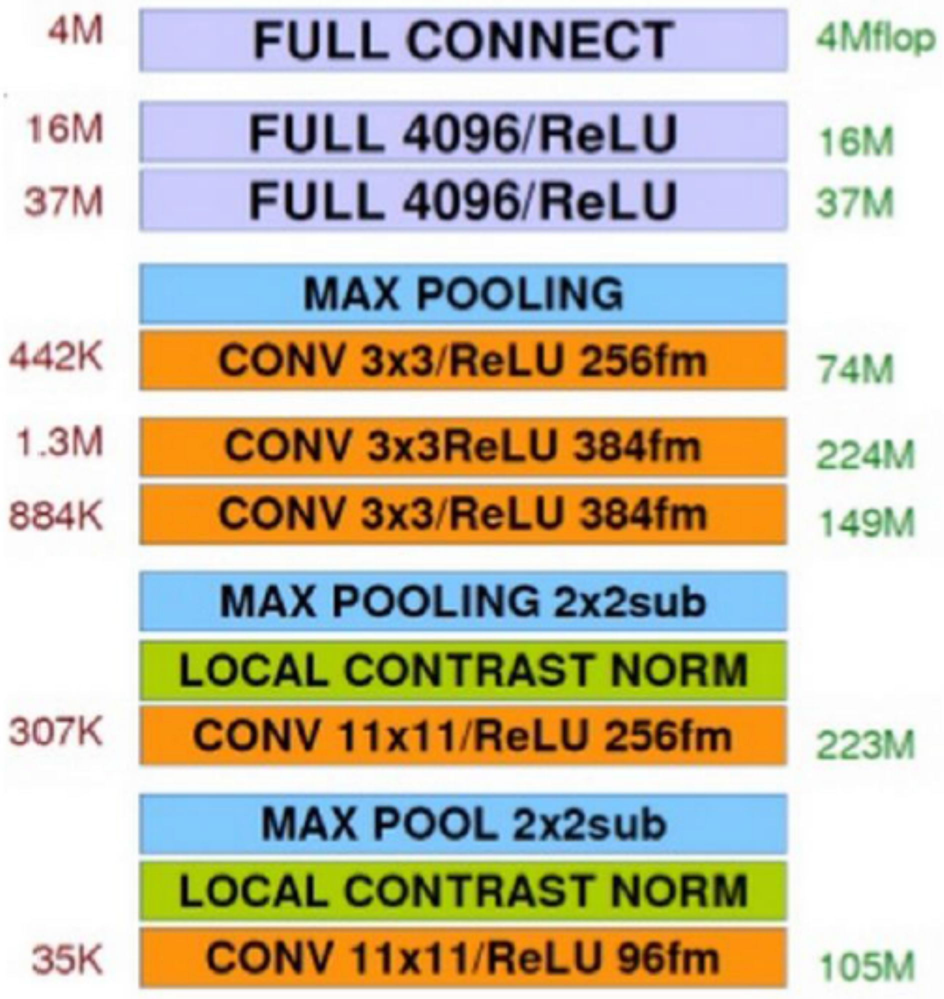
ImageNet 1K Competition (Fall 2012)



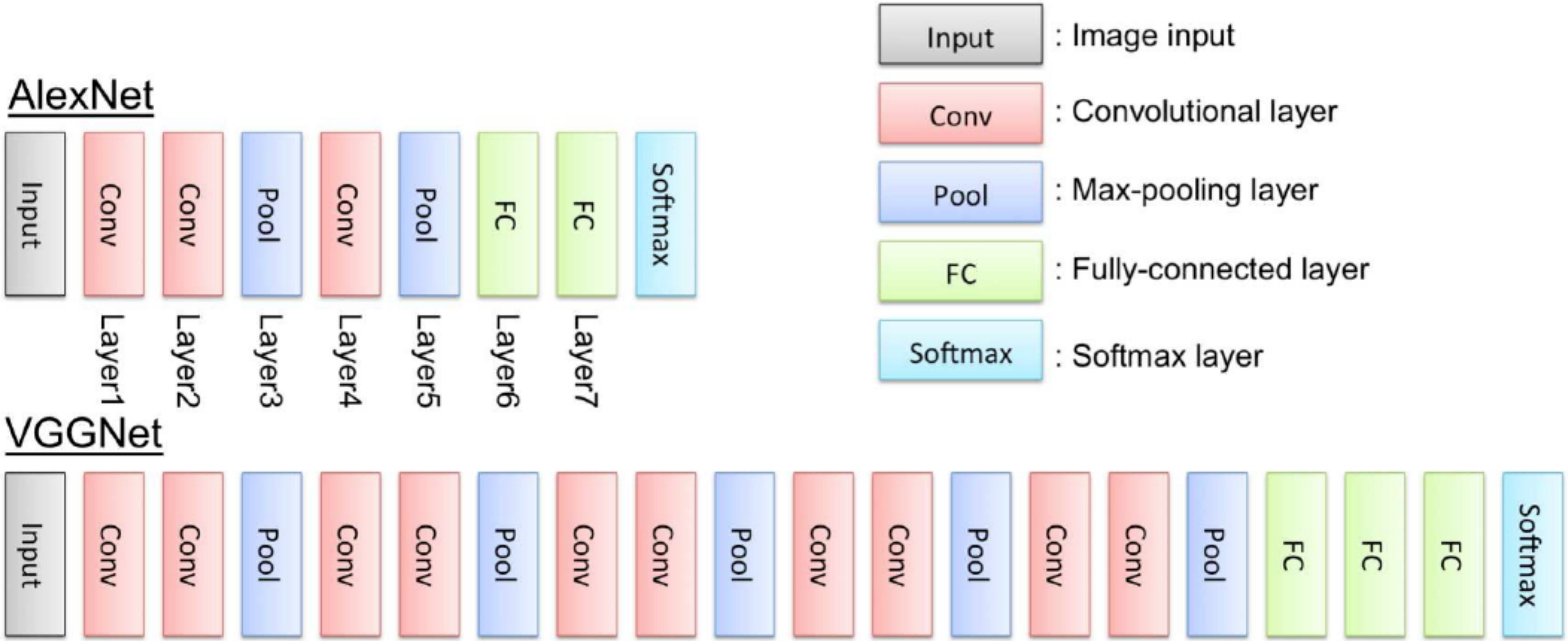
ImageNet 1K Competition (Fall 2013)



2012: The AlexNet

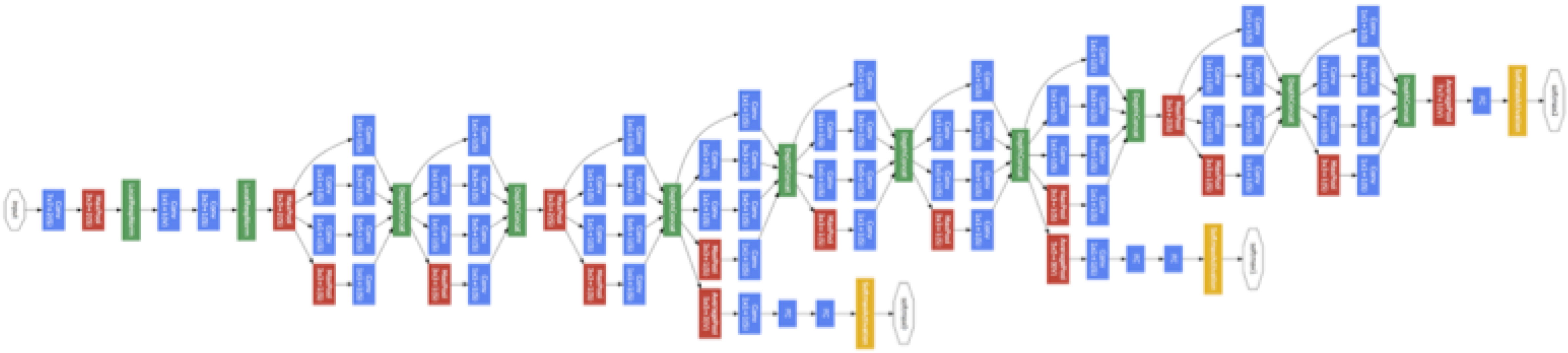


2013: VGGNet



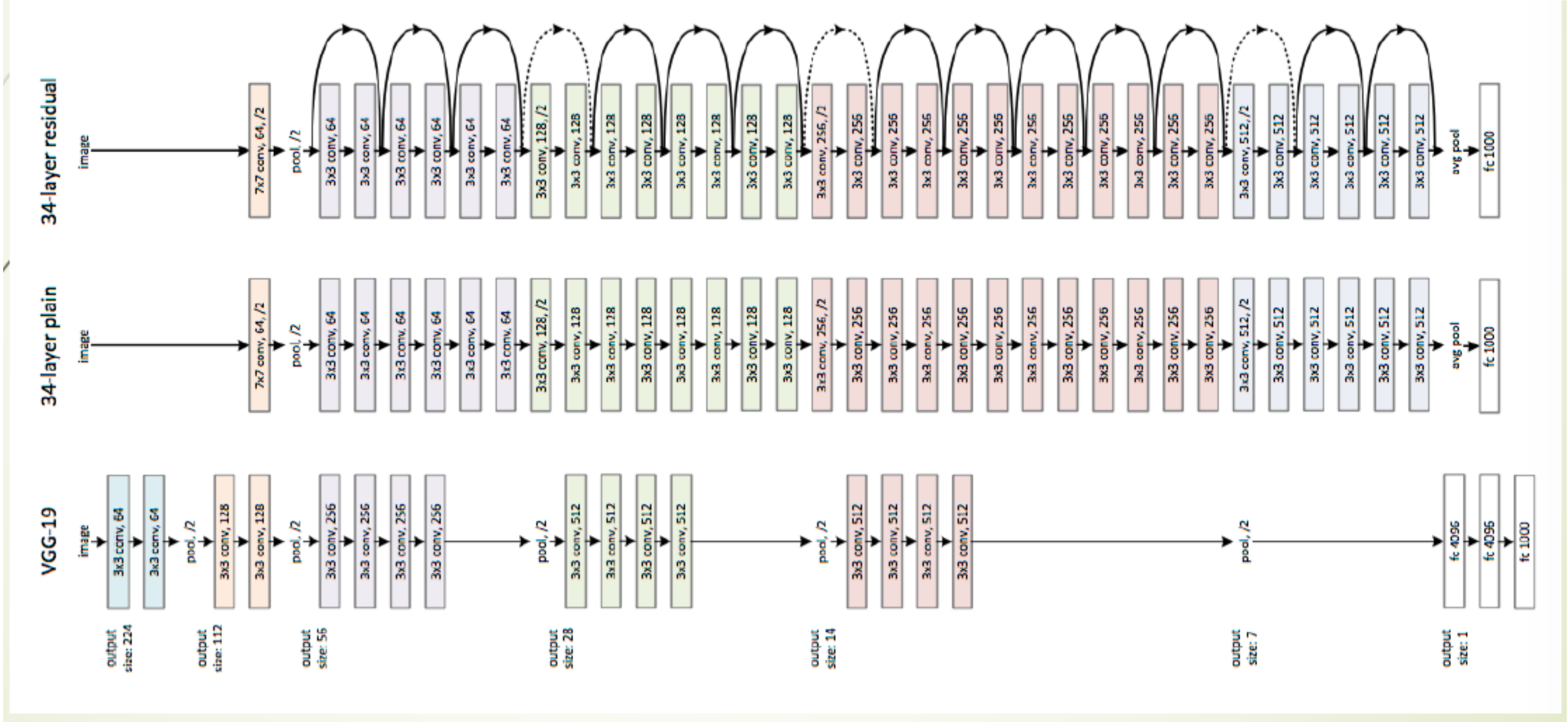
AlexNet
Input
Conv
Conv
Pool
Conv
Pool
FC
FC
Softmax
Layer1
Layer2
Layer3
Layer4
Layer5
Layer6
Layer7
VGGNet
Input
Conv
Conv
Pool
Conv
Conv
Pool
Conv
Conv
Pool
Conv
Conv
Pool
Conv
Conv
Pool
FC
FC
FC
Softmax
Input : Image input
Conv : Convolutional layer
Pool : Max-pooling layer
FC : Fully-connected layer
Softmax : Softmax layer

2014: GoogLeNet

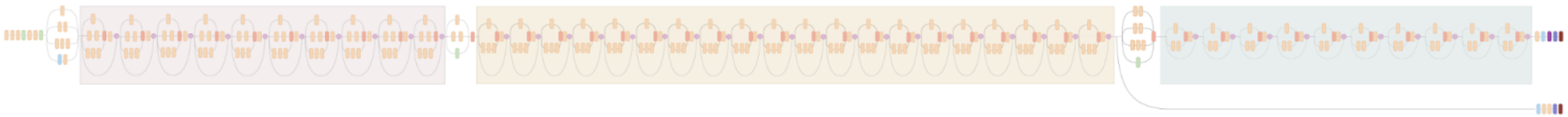


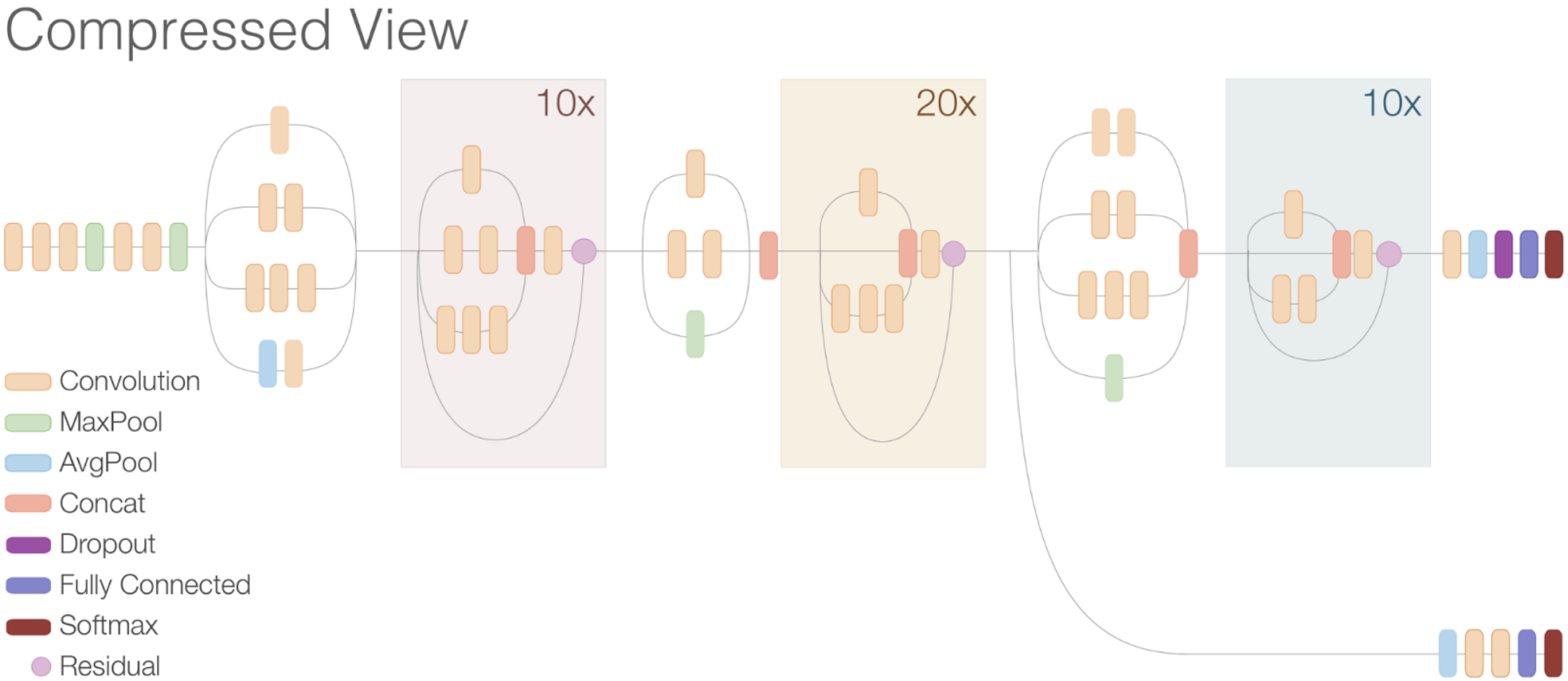
Convolution
Pooling
Softmax
Other

2015: ResNet

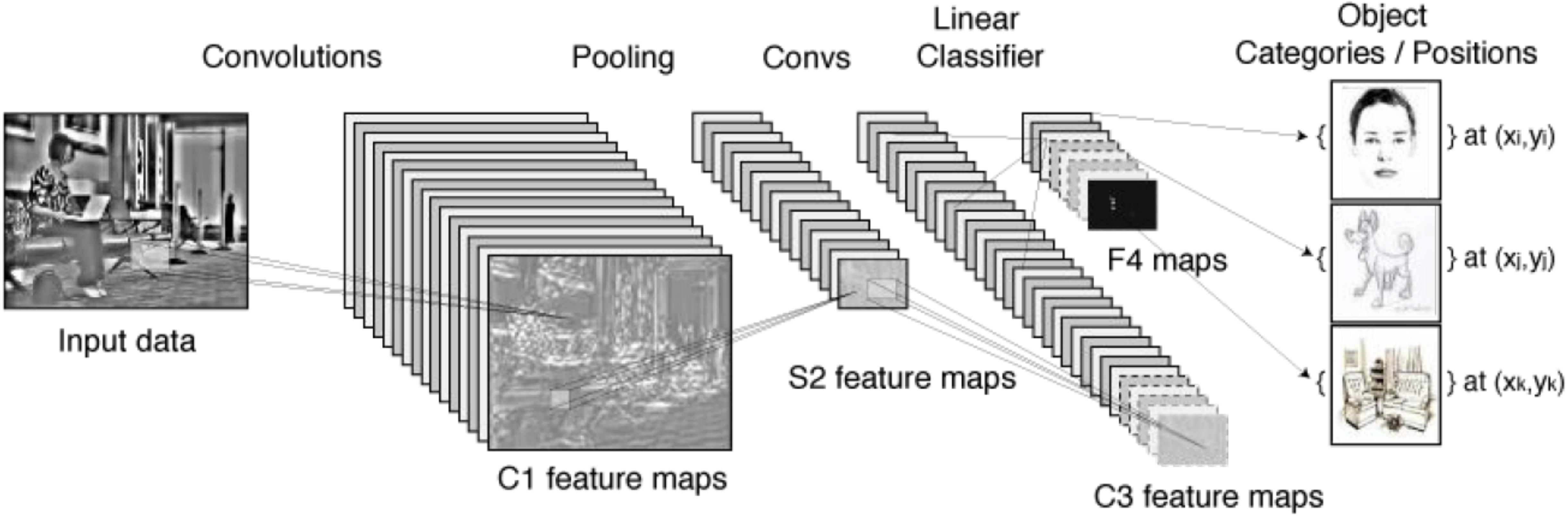

34-layer residual
34-layer plain
VGG-19
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 512, /2
3x3 conv, 512
avg pool
fc 1000
fc 4096
output size: 224
output size: 112
output size: 56
output size: 28
output size: 14
output size: 7
output size: 1

2016: Inception ResNet V2



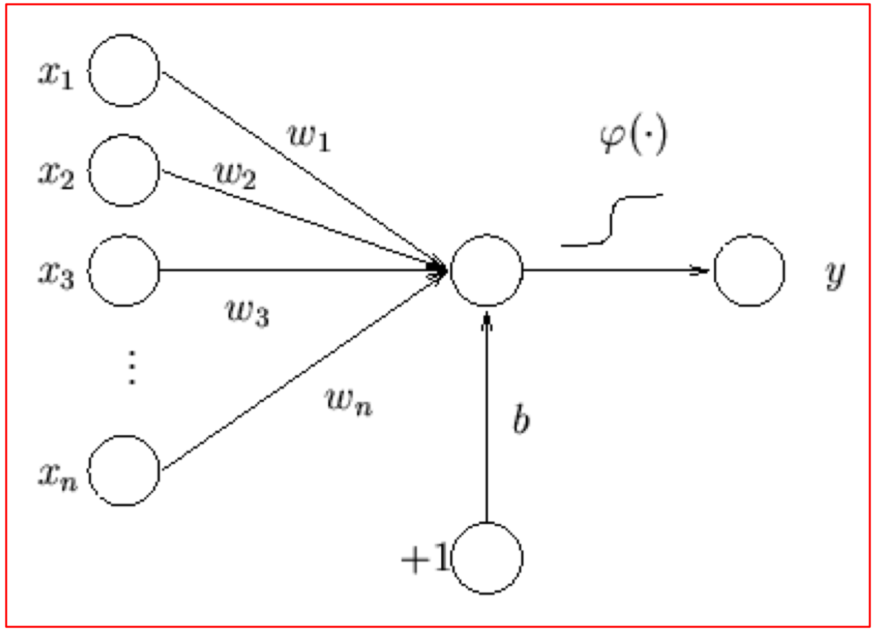


Convolutional Deep Neural Network (DNN)



ConvNet diagram from [Torch Tutorial]

An artificial neuron: **perceptron**



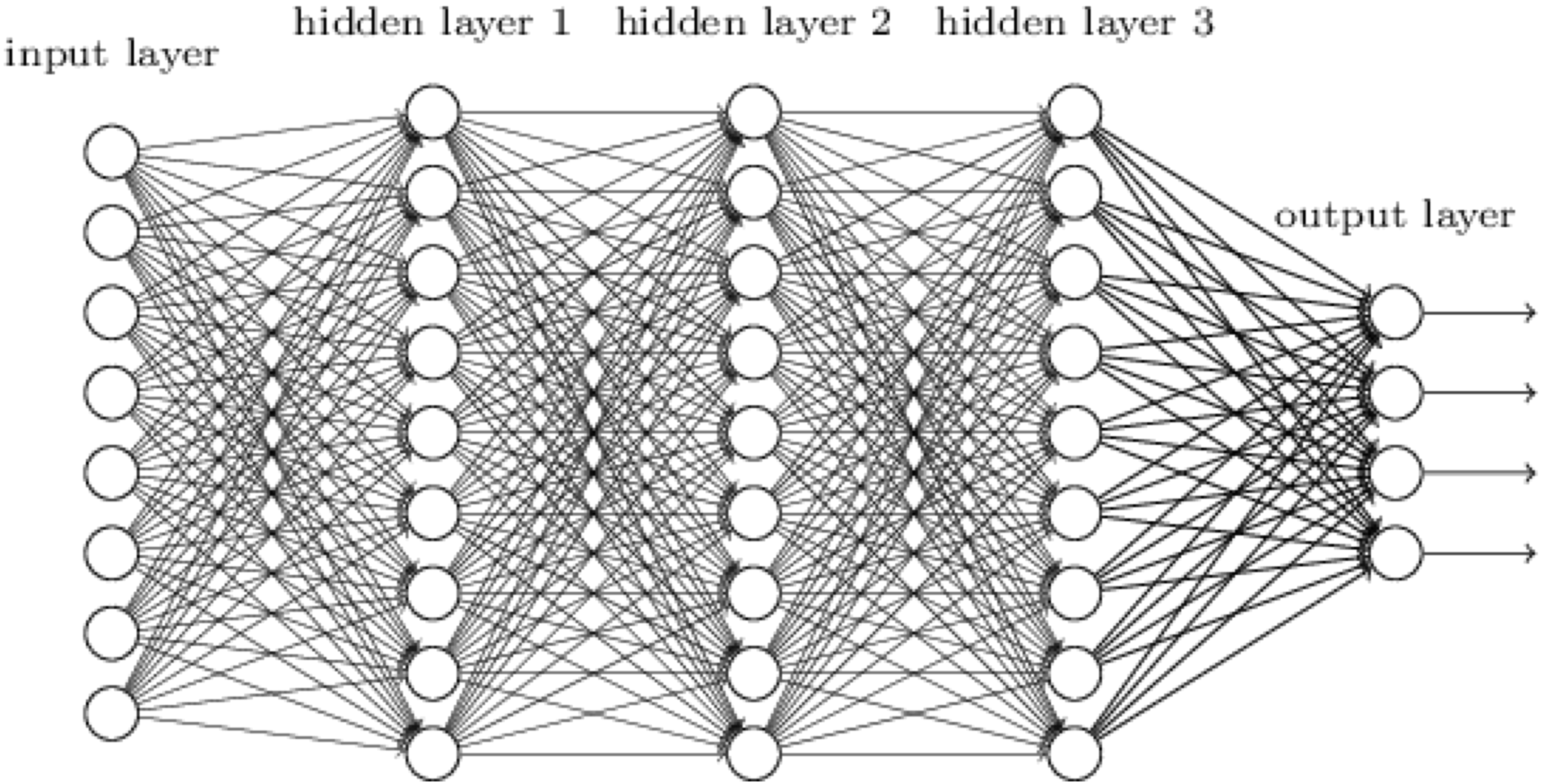
$$y = \varphi\left(\sum_{i=1}^{n} w_i x_i + b\right)$$
$$= \varphi(\mathbf{w}^T \mathbf{x} + b)$$

$\boxed{\varphi(\cdot) : \text{activation function}}$

- Introduced by Rosenblatt in 1958
- The basic building block for almost all DNNs

Image credit: www.hiit.fi/u/ahonkela/dippa/node41.html

Feedforward DNN: stacks of the perceprons



Learning the weights of DNNs

$$\Theta^\star \leftarrow \arg\min_{\Theta} \quad \mathbb{E}_{(x,y)\sim P_{XY}}\left[\mathrm{NET}(x;\Theta) \neq y\right]$$

Learning the weights of DNNs

$$\Theta^\star \leftarrow \arg\min_{\Theta} \quad \mathbb{E}_{(x,y)\sim P_{XY}}[\mathrm{NET}(x;\Theta) \neq y]$$

$$\hat{\Theta} \leftarrow \arg\min_{\Theta} \quad \frac{1}{n}\sum_{i=1}^{n}[\mathrm{NET}(x_i;\Theta) \neq y_i]$$

$$\hat{\Theta} \to \Theta^\star \text{ given many training data } (x_i, y_i), i = 1, 2, \cdots, n$$

Pros and cons

Very flexible	
Easy to use	
Easy to configure	
Easy to train with off-shelf tools	

Pros and cons

Very flexible	Very flexible perhaps too much
Easy to use	Hard to understand
Easy to configure	Hard to configure
Easy to train with off-shelf tools	Hard to train (overfitting, not robust)

Dropout

$$\hat{\Theta} \leftarrow \arg\min_{\Theta} \quad \frac{1}{n}\sum_{i=1}^{n}[\mathrm{NET}(x_i;\Theta) \neq y_i]$$

[Srivastava et al.; JMLR'14]

- Set a neuron to 0 with p=0.5
- Enforce survived neurons to learn; reduce co-adaptation

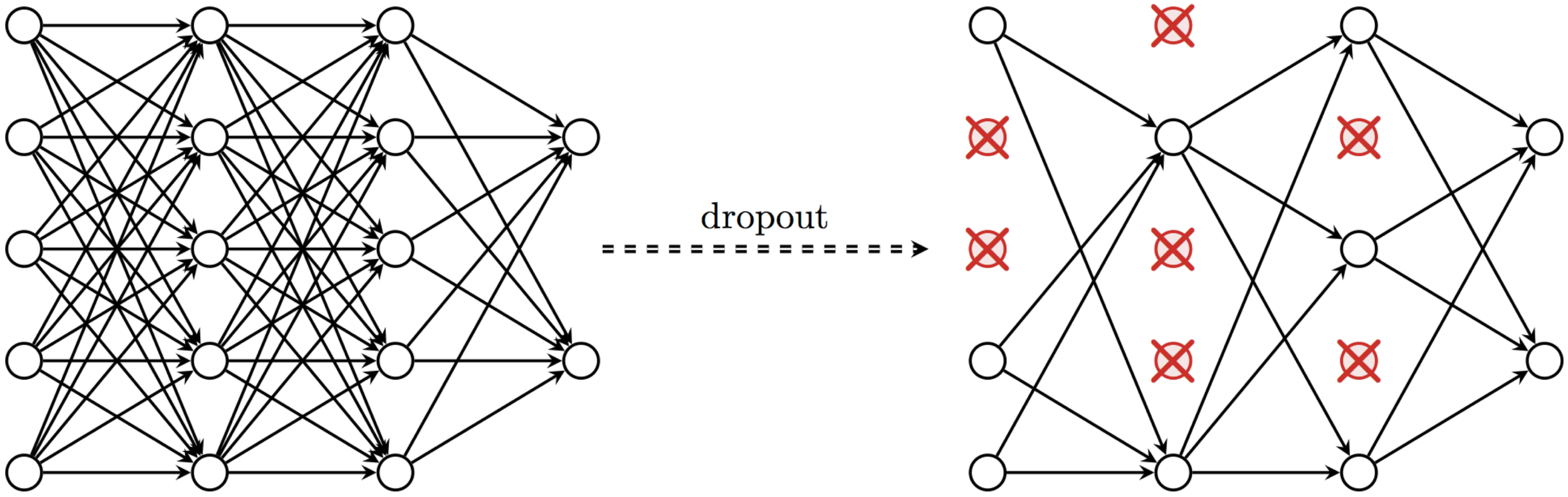


Image credit: https://github.com/PetarV-/TikZ/tree/master/Dropout

Multinomial dropout

- Let neurons compete with each other [Li, Gong, Yang; NIPS'16]
- Dropout rates follow a multinomial distribution

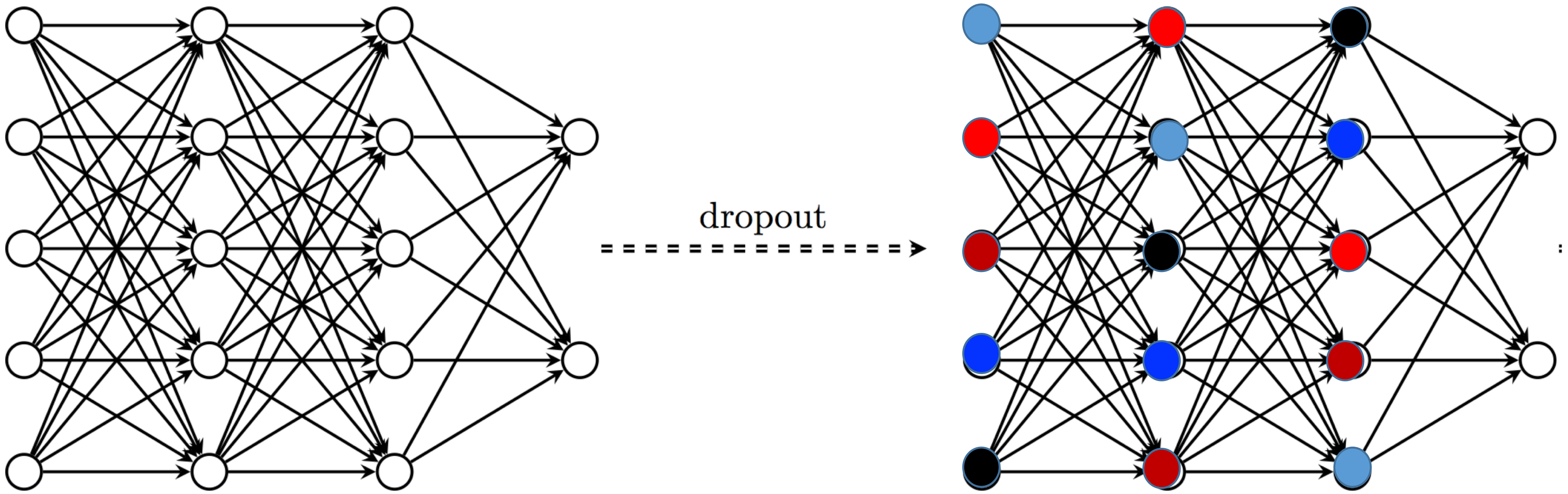


Image credit: https://github.com/PetarV-/TikZ/tree/master/Dropout

Multinomial dropout

- Let neurons compete with each other [Li, Gong, Yang; NIPS'16]
- Dropout rates follow a multinomial distribution

Theorem 1. *Let $\mathcal{L}(\mathbf{w})$ be the expected risk of $\mathbf{w}$ defined in (1). Assume $\mathrm{E}_{\widehat{\mathcal{D}}}[\|\mathbf{x} \circ \boldsymbol{\epsilon}\|_2^2] \leq B^2$ and $\ell(z, y)$ is G-Lipschitz continuous. For any $\|\mathbf{w}_*\|_2 \leq r$, by appropriately choosing η, we can have*

$$\mathrm{E}[\mathcal{L}(\widehat{\mathbf{w}}_n) + R_{\mathcal{D},\mathcal{M}}(\widehat{\mathbf{w}}_n)] \leq \mathcal{L}(\mathbf{w}_*) + R_{\mathcal{D},\mathcal{M}}(\mathbf{w}_*) + \frac{GBr}{\sqrt{n}}$$

where $\mathrm{E}[\cdot]$ is taking expectation over the randomness in $(\mathbf{x}_t, y_t, \boldsymbol{\epsilon}_t), t = 1, \ldots, n$.

Data-dependent multinomial dropout

- Let neurons compete with each other [Li, Gong, Yang; NIPS'16]
- Dropout rates follow a multinomial distribution

$$p_i = \frac{\sqrt{\frac{1}{n}\sum_{j=1}^{n}[[\mathbf{x}_j]_i^2]}}{\sum_{i'=1}^{d}\sqrt{\frac{1}{n}\sum_{j=1}^{n}[[\mathbf{x}_j]_{i'}^2]}}$$

- Neurons of higher "variance" --- larger weights

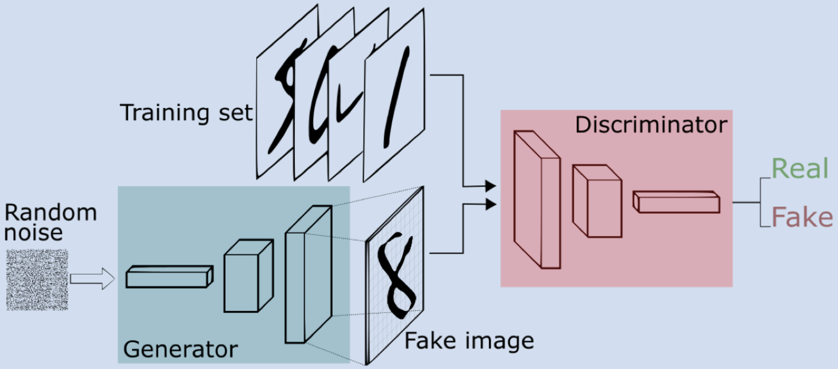
Multiple Shades of Dropout for Discriminative and *Generative* Deep Neural Networks

[1] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958.

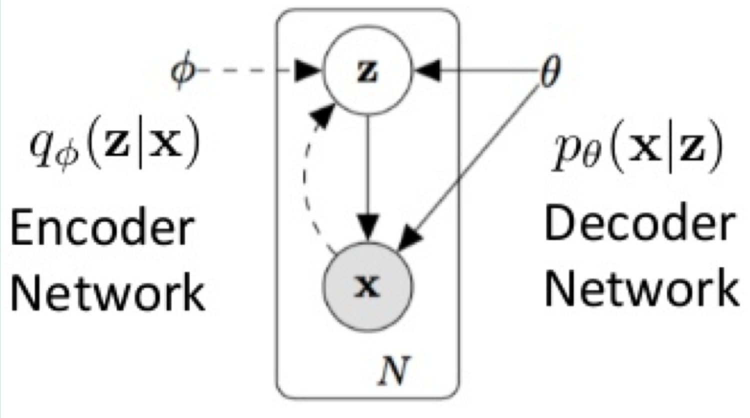
[2] Li, Z., Gong, B., & Yang, T. (2016). Improved dropout for shallow and deep learning. In *Advances in Neural Information Processing Systems* (pp. 2523-2531).

Generative DNNs

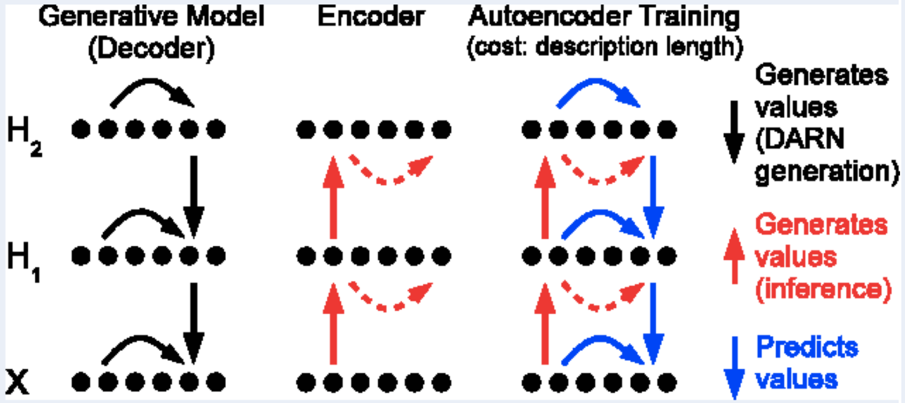
Generative adversarial net



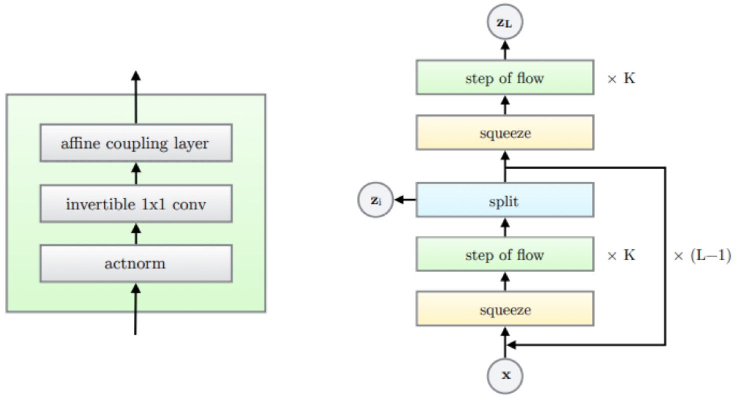
Variational auto-encoder



Autoregressive model



GLOW



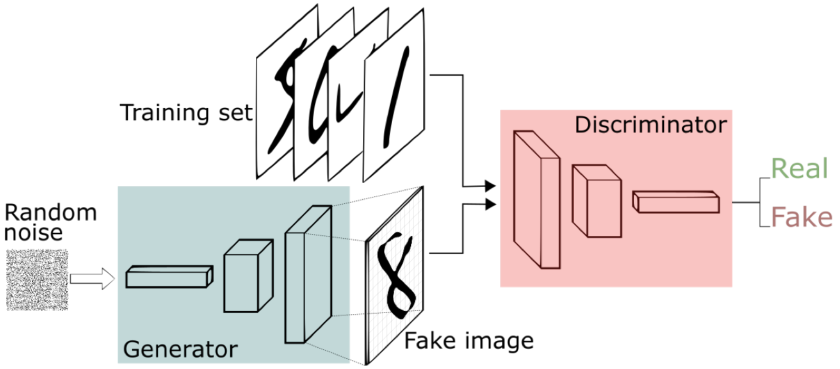
(a) One step of our flow.

(b) Multi-scale architecture (Dinh et al., 2016).

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

Generative DNNs

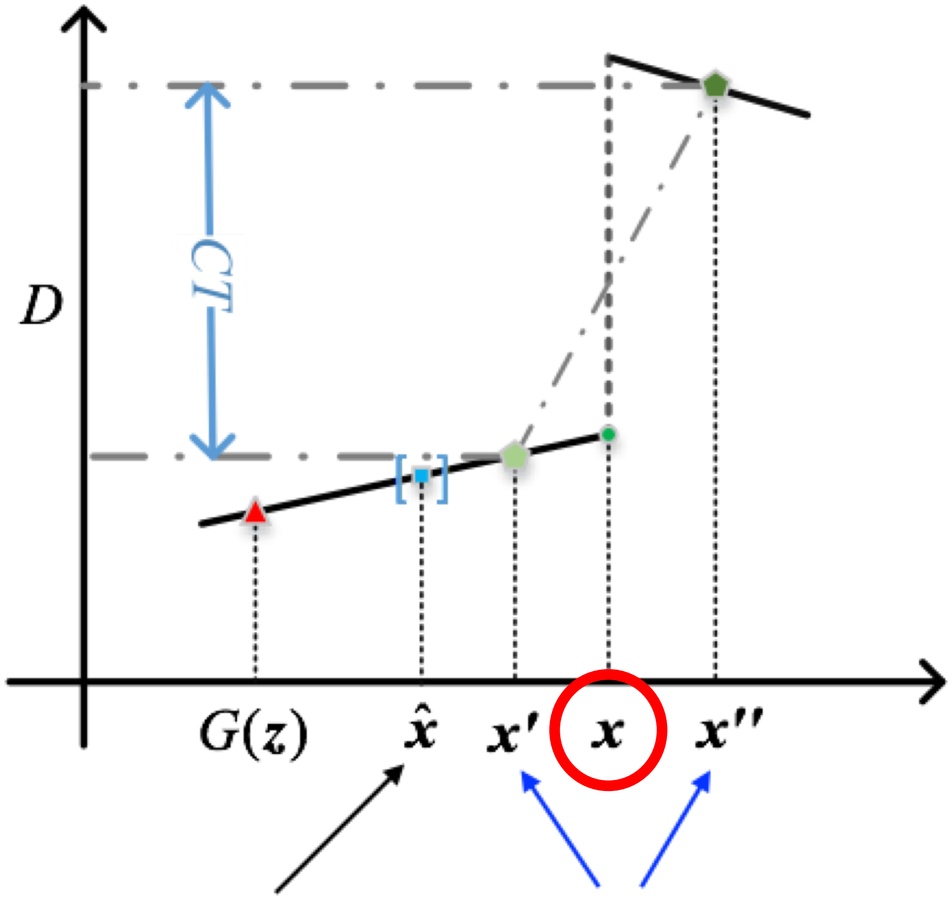
Generative adversarial net



$$\frac{d(f(\boldsymbol{x}'), f(\boldsymbol{x}''))}{d(\boldsymbol{x}', \boldsymbol{x}'')} \leq 1$$

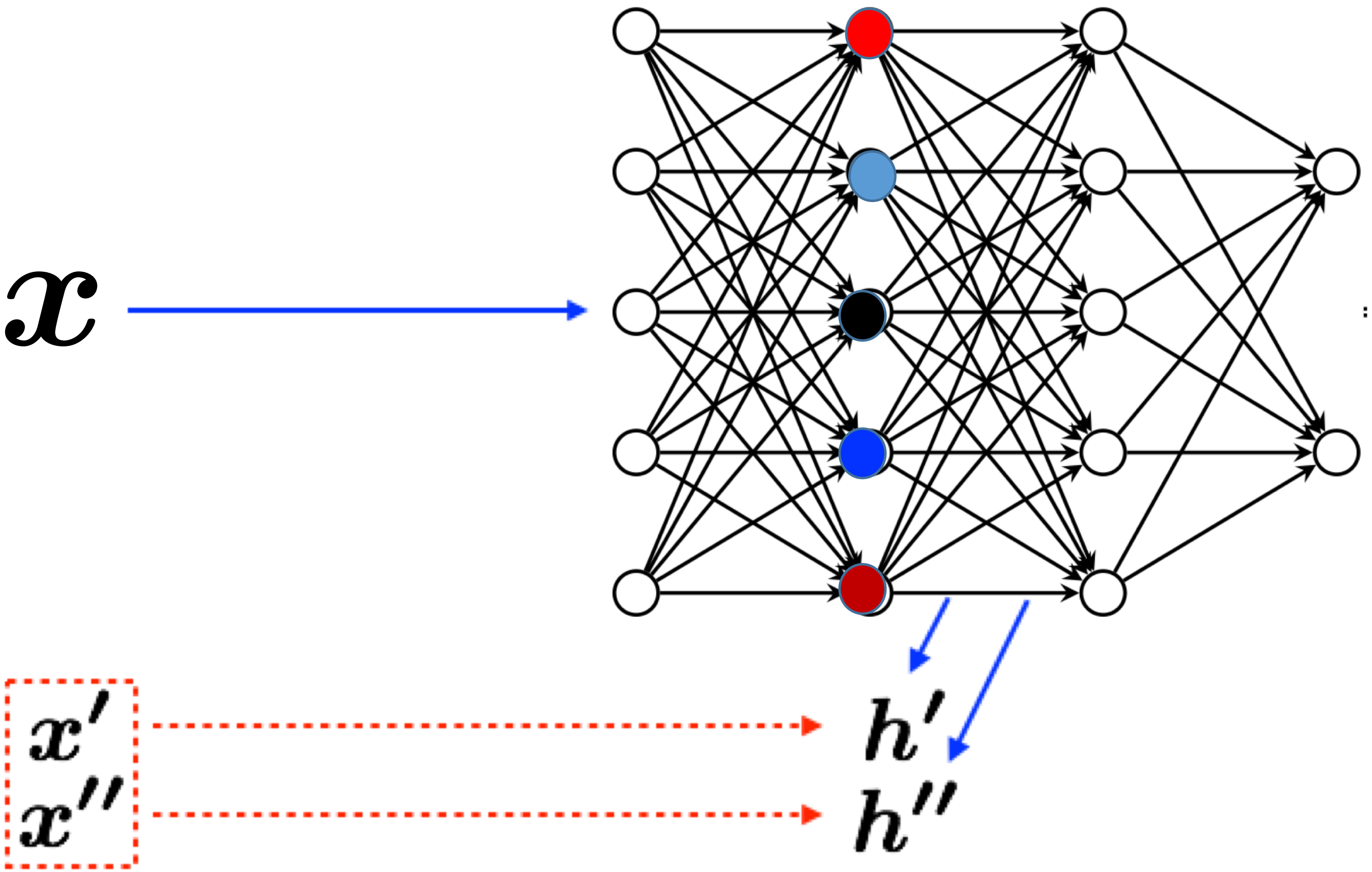
Image credit: https://skymind.ai/wiki/generative-adversarial-network-gan

Generative adversarial net (GAN)



[Gulrajani et al., NIPS'17] [Wei, Gong, et al., ICLR'18]

Generative adversarial net (GAN)



[1] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958.

[2] Li, Z., Gong, B., & Yang, T. (2016). Improved dropout for shallow and deep learning. In *Advances in Neural Information Processing Systems* (pp. 2523-2531).

[3] Wei, X., Gong, B., Liu, Z., Lu, W., & Wang, L. (2018). Improving the Improved Training of Wasserstein GANs: A Consistency Term and Its Dual Effect. *International Conference on Learning Representations*.

Take-home message

Dropout

Independently drops some neurons in training

Effectively prevents network overfitting

Data-dependent multinomial dropout

Lets neurons compete with other

Attends more on neurons of larger “variance”

Yields provably better generalization bound

Dropout for (Wasserstein) GAN

Dropout twice per input → Lipschitz continuity