Object Recognition – Part 2

Danna Gurari
University of Colorado Boulder
Fall 2023

CC BY
https://home.cs.colorado.edu/~DrG/Courses/RecentAdvancesInComputerVision/AboutCourse.html

Review

- Last lecture:
  - Object recognition problem
  - Object recognition applications
  - Object recognition datasets
  - Object recognition evaluation metric
  - A popular solution: convolutional neural networks

- Assignments (Canvas)
  - Reading assignment was due earlier today
  - Next reading assignments due Wednesday and next Monday

- Questions?

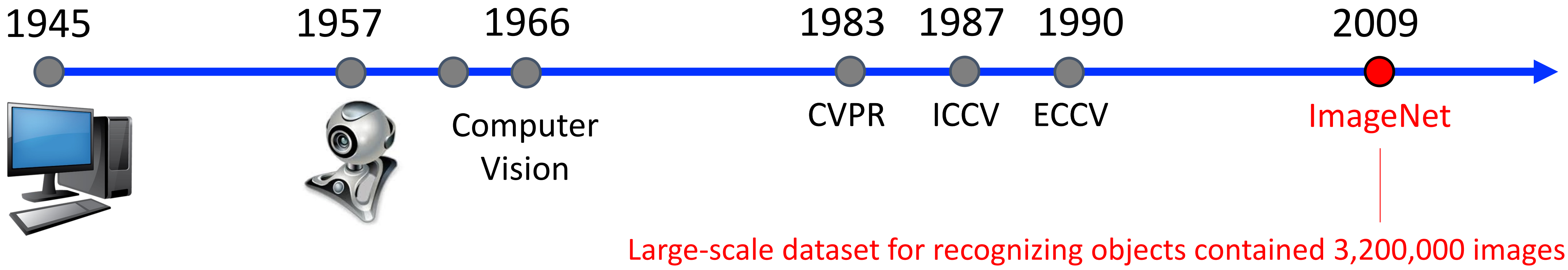
Object Recognition: Today’s Topics

- ImageNet Challenge Top Performers
- Baseline Model: AlexNet
- VGG
- ResNet
- Discussion

Object Recognition: Today's Topics

- ImageNet Challenge Top Performers
- Baseline Model: AlexNet
- VGG
- ResNet
- Discussion

Recall: Catalyst for Computer Vision Revolution



J. Deng, W. Dong, R. Socher, L. Li, K. Li and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. 2009.

Recall: Catalyst for Computer Vision Revolution

Progress of models on ImageNet (Top 5 Error)

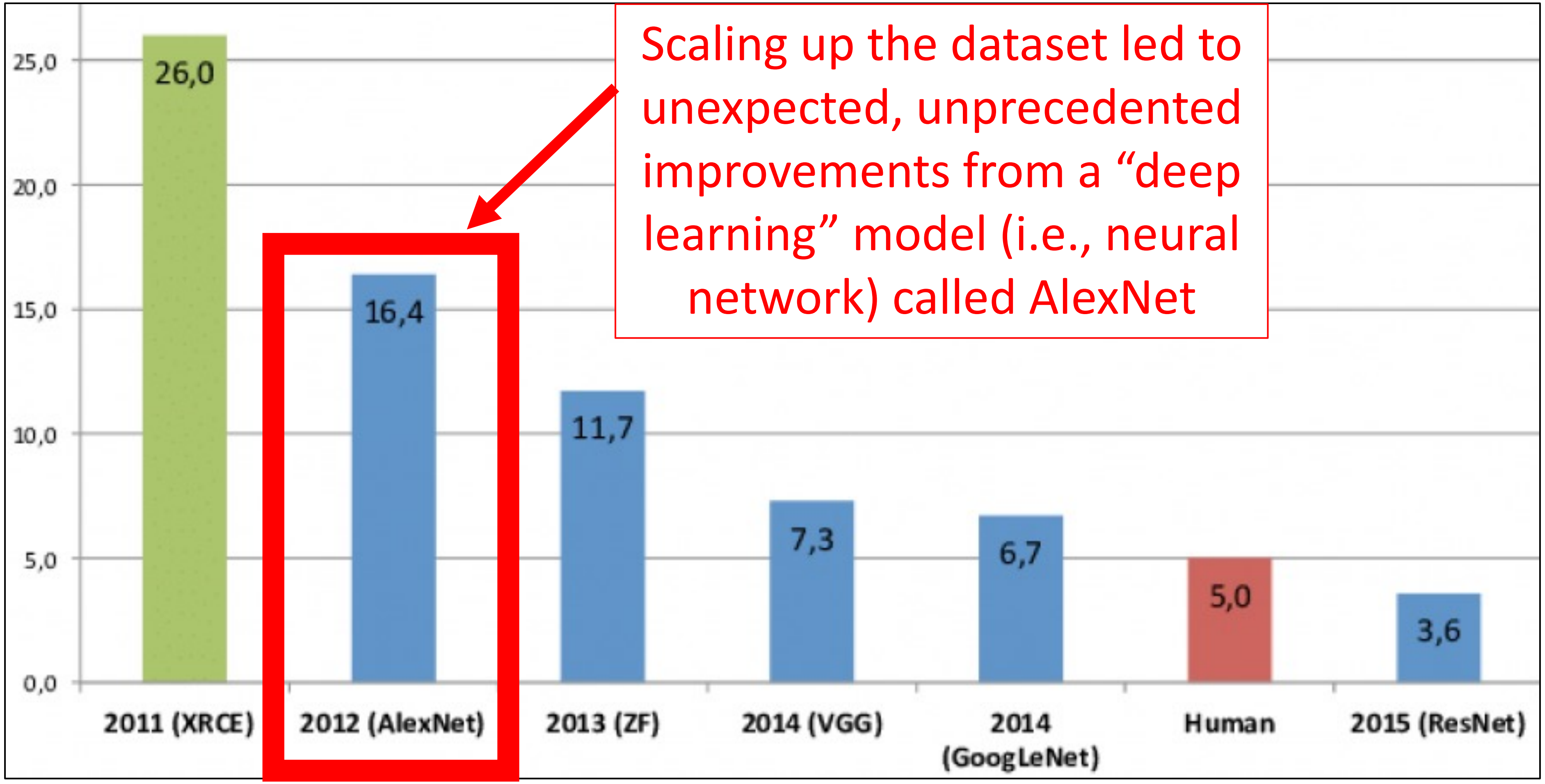
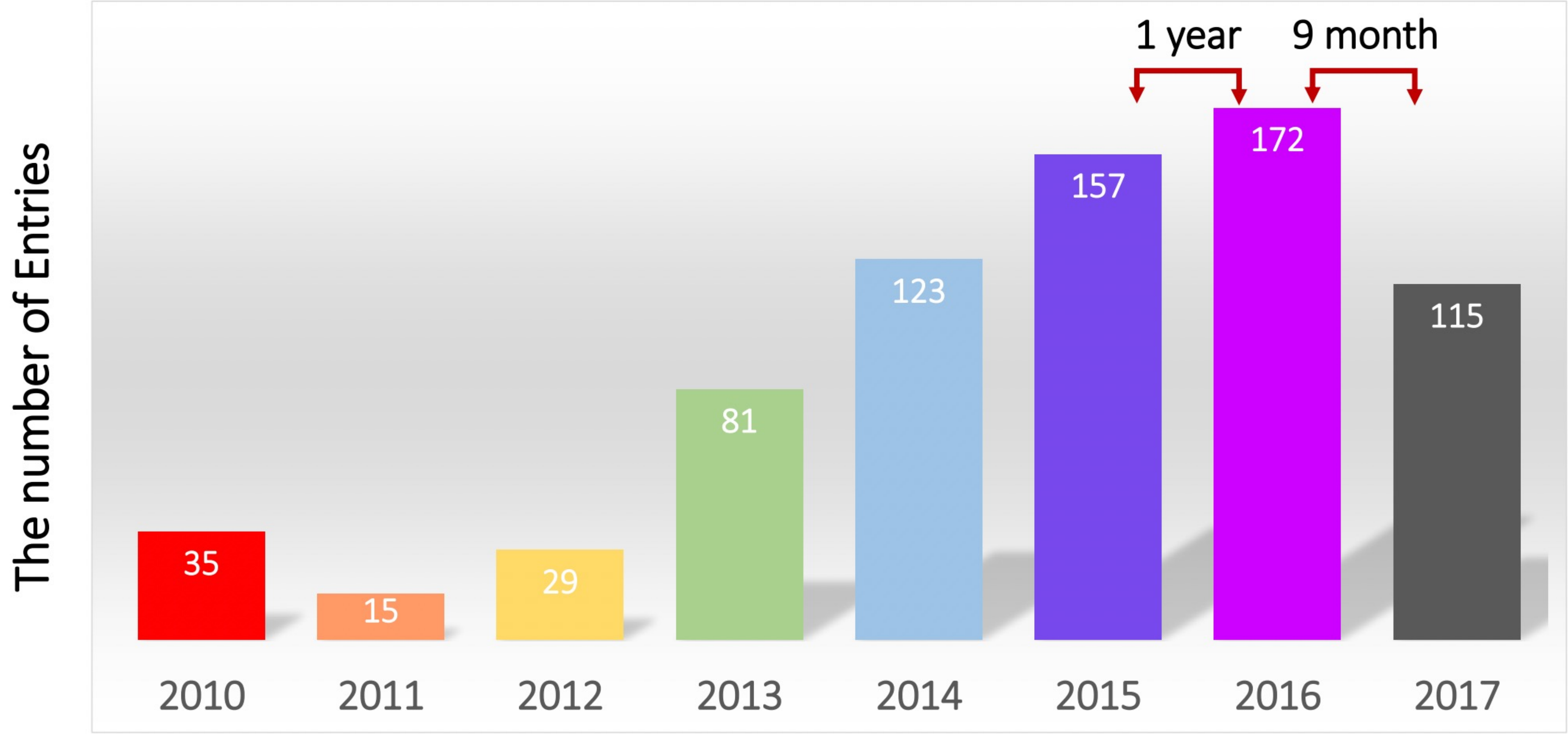


Figure Source: https://www.edge-ai-vision.com/2018/07/deep-learning-in-five-and-a-half-minutes/

Recall: ImageNet Challenge Submissions



Inspired by AlexNet, many more researchers in the computer vision community proposed neural networks and showed how to make further progress over the years!

Source: https://image-net.org/static_files/files/ILSVRC2017_overview.pdf

Secret Sauce for State-of-Art: Deeper CNNs

Progress of models on ImageNet (Top 5 Error)

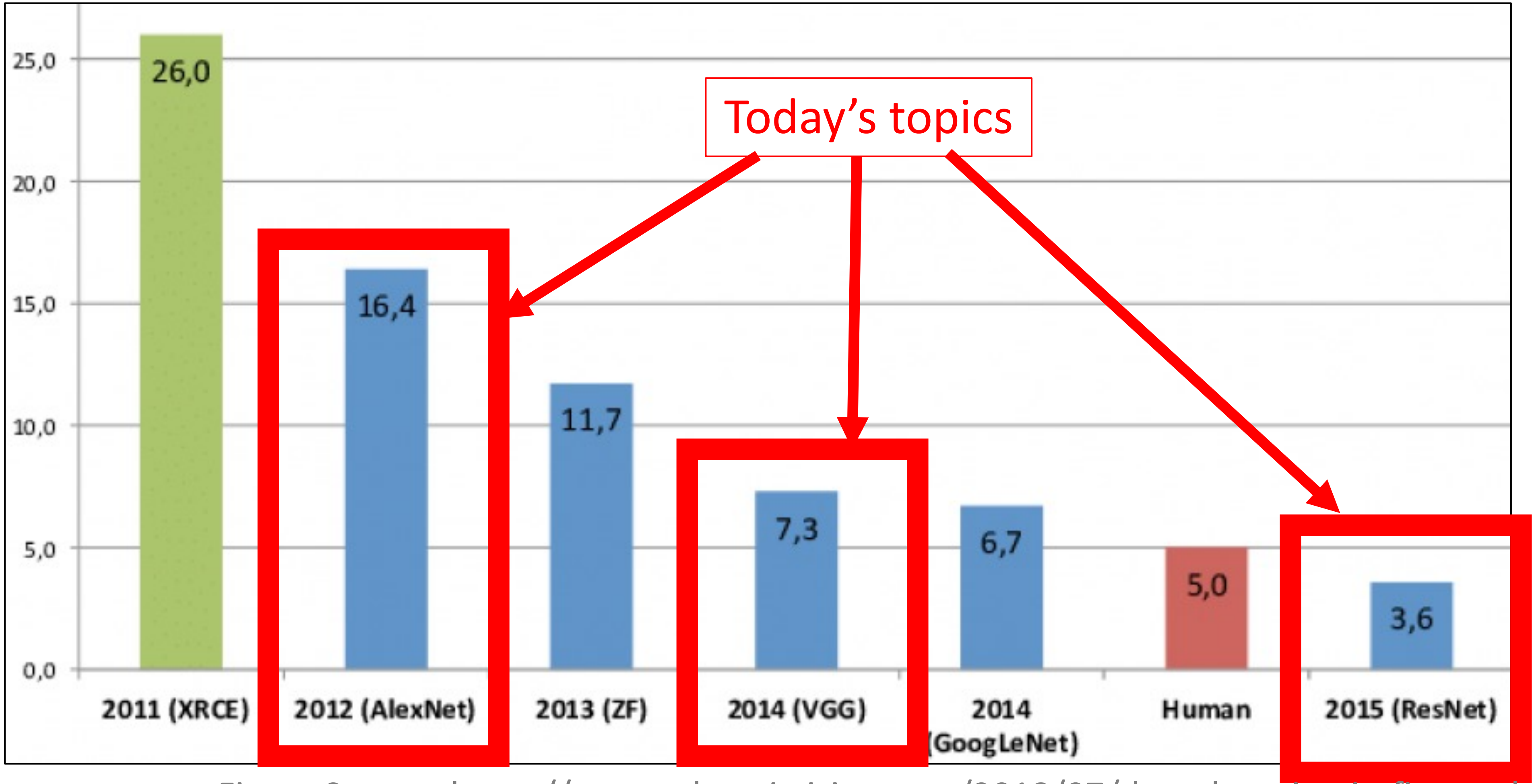
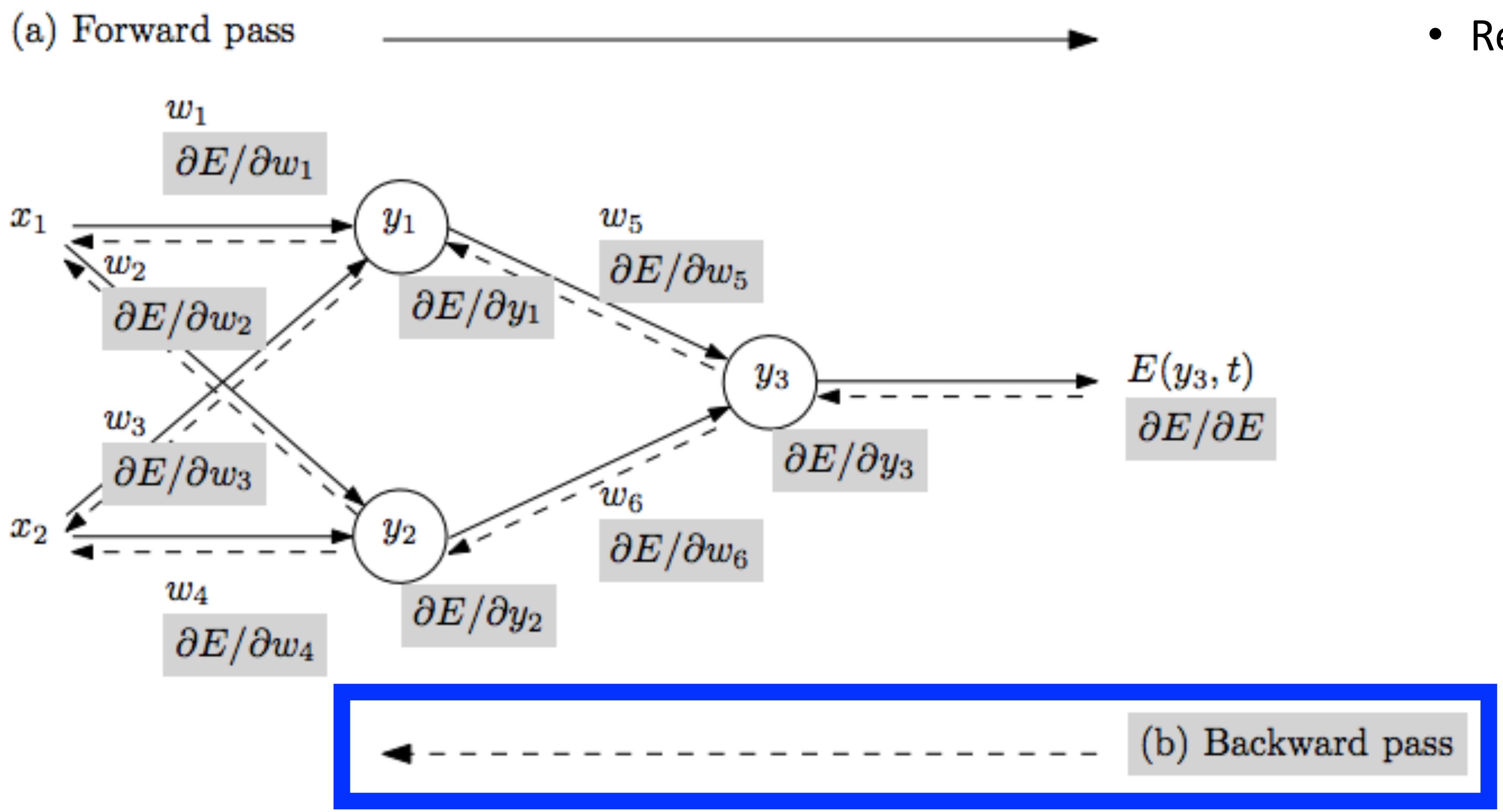


Figure Source: https://www.edge-ai-vision.com/2018/07/deep-learning-in-five-and-a-half-minutes/

Why It Is Difficult to Achieve Better Performance with CNNs That Are Deeper: Vanishing Gradients



$$W_x = W_x - \alpha\left(\frac{\partial Error}{\partial W_x}\right)$$

- Repeat until stopping criterion met:
  1. **Forward pass**: propagate training data through model to make predictions
  2. **Error quantification**: measure dissatisfaction with a model's predictions on training data
  3. **Backward pass**: using predicted output, calculate gradients backward to assign blame to each model parameter
  4. Update each parameter using calculated gradients

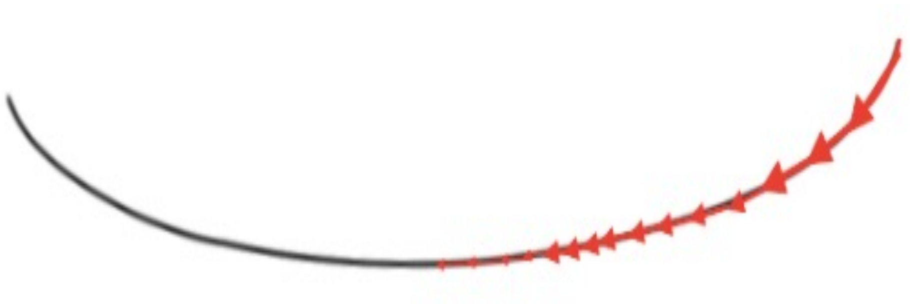
https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/

Why It Is Difficult to Achieve Better Performance with CNNs That Are Deeper: Vanishing Gradients

- **Want**: objective function with a gradient large enough to support (efficient) learning; e.g.,

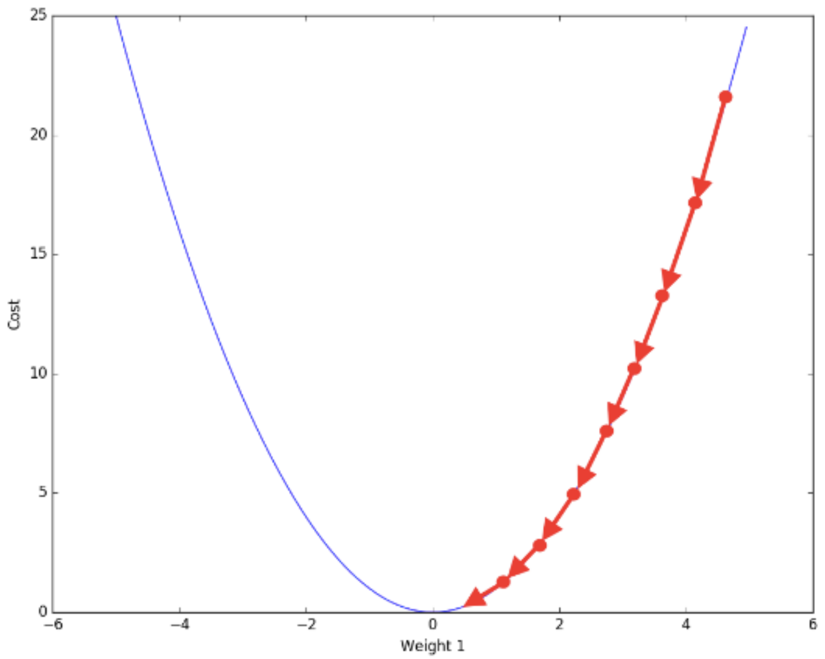
Problem

Function 1



Small gradients so parameters change slowly

Function 2



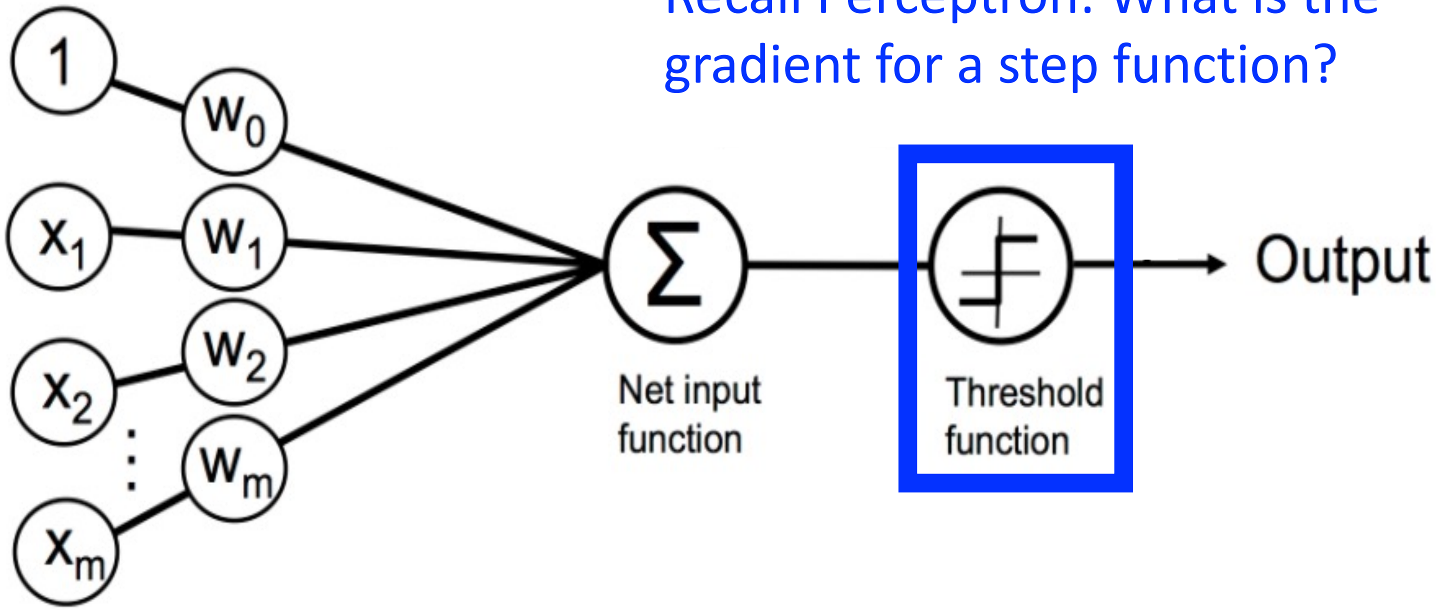
Large gradients so parameters change quickly

$$W_x = W_x - \alpha \boxed{\left(\frac{\partial Error}{\partial W_x}\right)}$$

https://ayearofai.com/rohan-4-the-vanishing-gradient-problem-ec68f76ffb9b

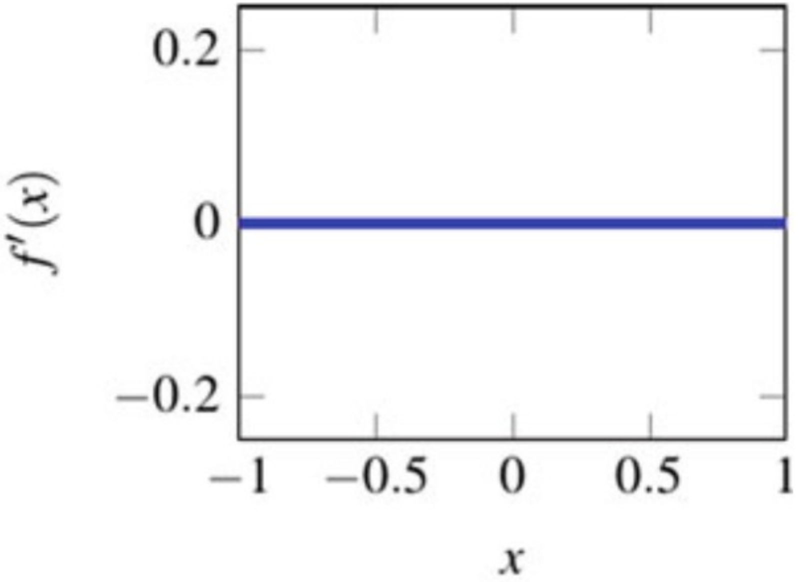
Why It Is Difficult to Achieve Better Performance with CNNs That Are Deeper: Vanishing Gradients

Recall Perceptron: What is the gradient for a step function?



Python Machine Learning; Raschka & Mirjalili

0 everywhere except 0 where it is non-differentiable



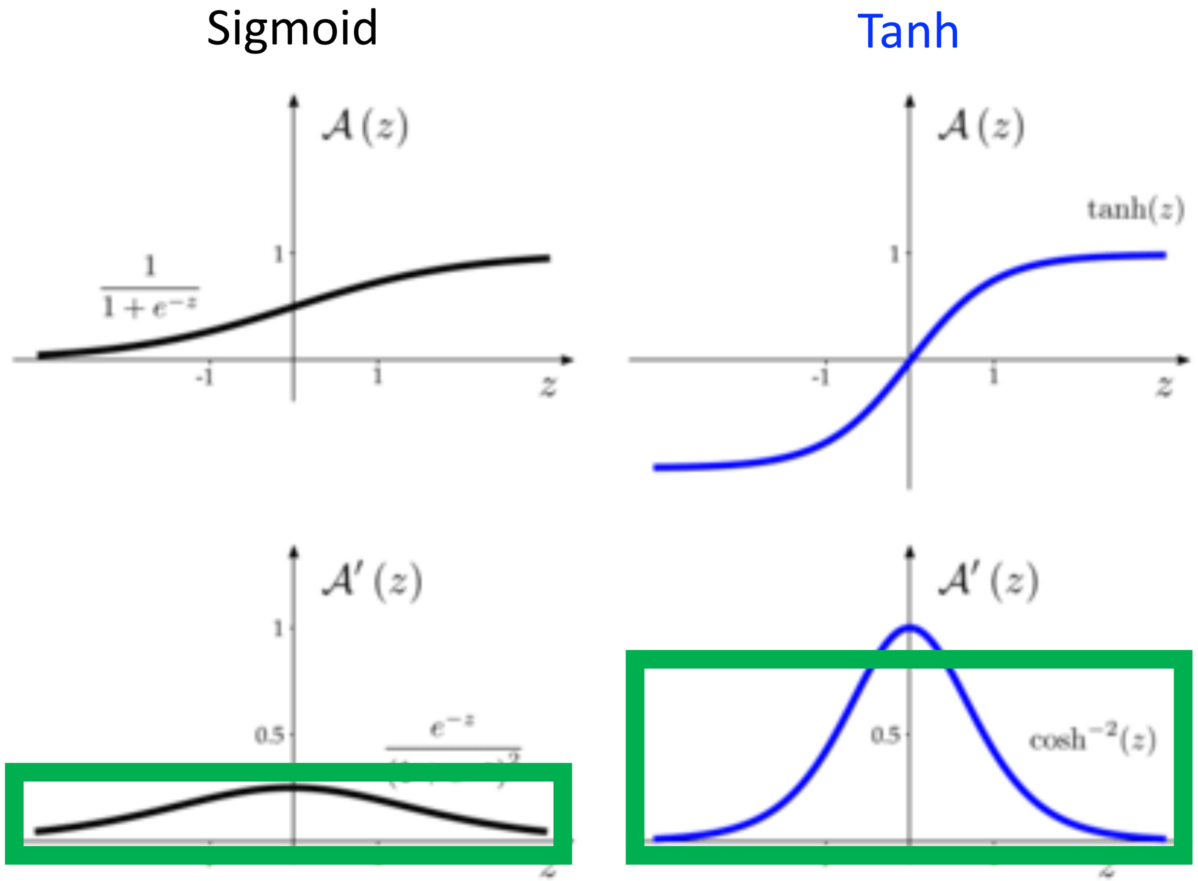
Deep Learning for NLP and Speech Recognition; Kamath, Liu, & Whitaker

No gradient means model parameters wouldn't change with gradient descent!

Why It Is Difficult to Achieve Better Performance with CNNs That Are Deeper: Vanishing Gradients

Recall: Activation Functions

Small gradients limit amount model parameters change with gradient descent

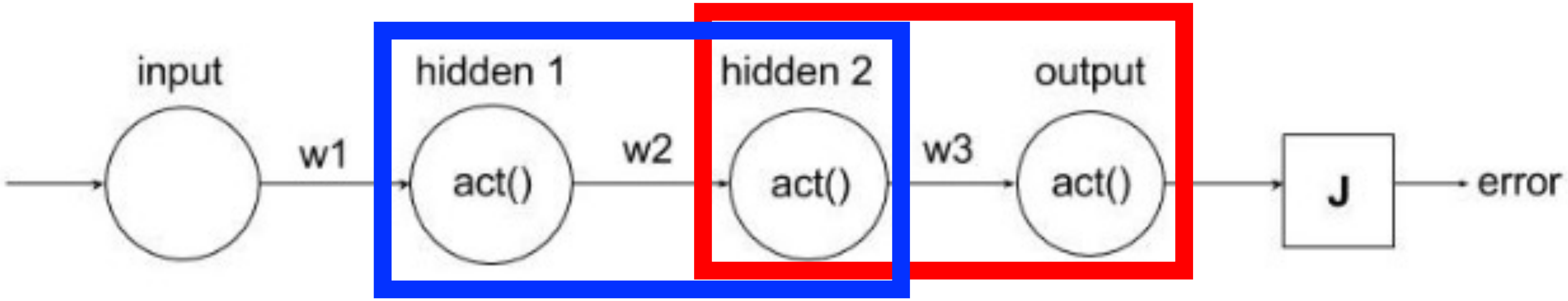


$$W_x = W_x - \alpha \boxed{\left(\frac{\partial Error}{\partial W_x}\right)}$$

Masi et al. Journal of the Mechanics and Physics of Solids. 2021

Why It Is Difficult to Achieve Better Performance with CNNs That Are Deeper: Vanishing Gradients

- Toy example:



- Error Derivative with respect to weight w1:

$$\frac{\partial error}{\partial w1} = \frac{\partial error}{\partial output} * \boxed{\frac{\partial output}{\partial hidden2}} * \boxed{\frac{\partial hidden2}{\partial hidden1}} * \frac{\partial hidden1}{\partial w1}$$

e.g., derivative of sigmoid activation function: (0 to 1/4]

e.g., derivative of sigmoid activation function: (0 to 1/4]

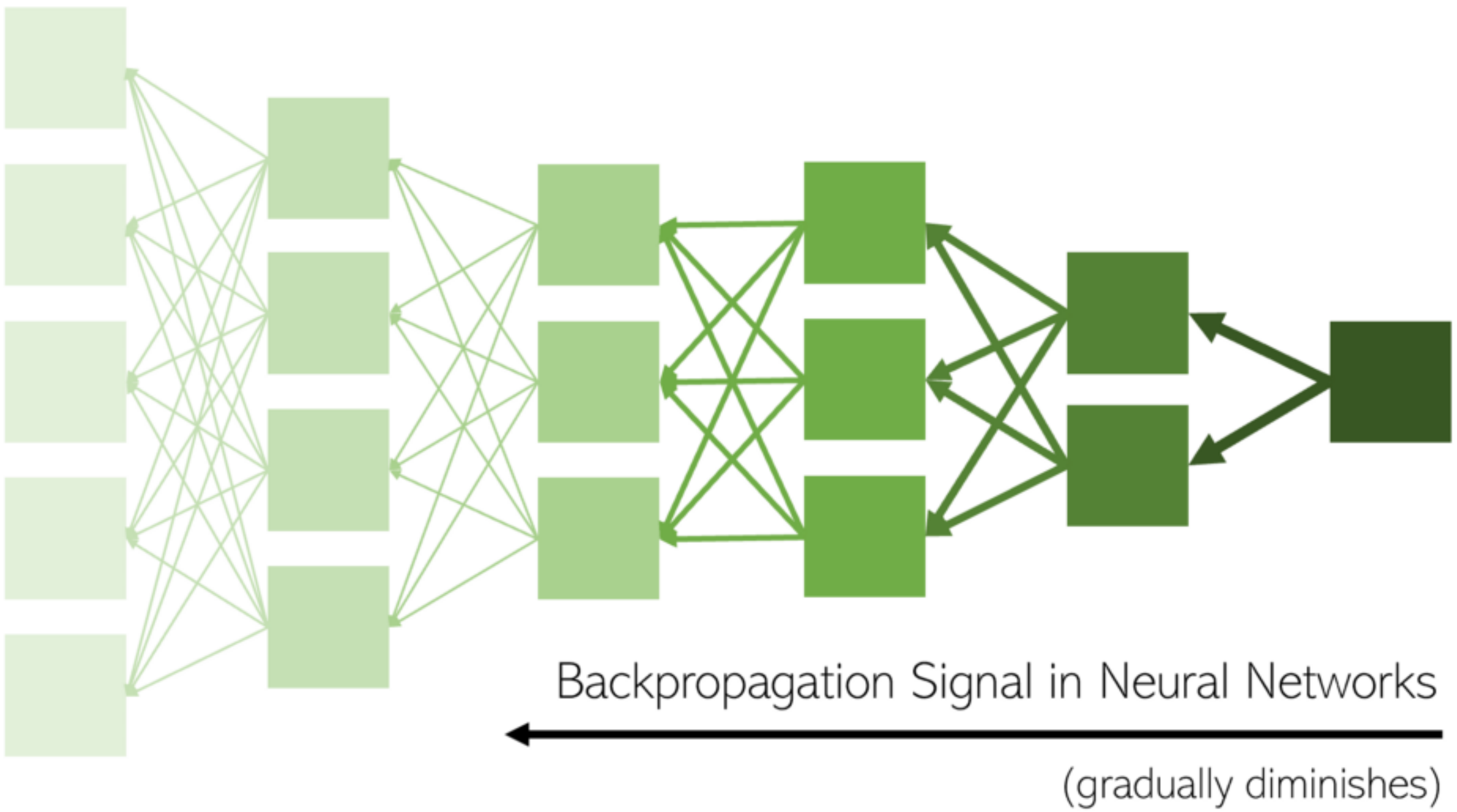
$$W_x = W_x - \alpha \boxed{\left(\frac{\partial Error}{\partial W_x}\right)}$$

Problem: When multiplying more numbers smaller than 1, gradient decreases leading to reduced weight changes!

https://ayearofai.com/rohan-4-the-vanishing-gradient-problem-ec68f76ffb9b

Why It Is Difficult to Achieve Better Performance with CNNs That Are Deeper: Vanishing Gradients

Smallest gradients at **earliest layers make them slowest to train**, yet later layers depend on those earlier layers to do something useful; consequently, NNs struggle with garbage in means garbage out



https://towardsdatascience.com/batch-normalization-the-greatest-breakthrough-in-deep-learning-77e64909d81d

Object Recognition: Today’s Topics

- ImageNet Challenge Top Performers
- Baseline Model: AlexNet
- VGG
- ResNet
- Discussion

AlexNet: A Deeper CNN

Progress of models on ImageNet (Top 5 Error)

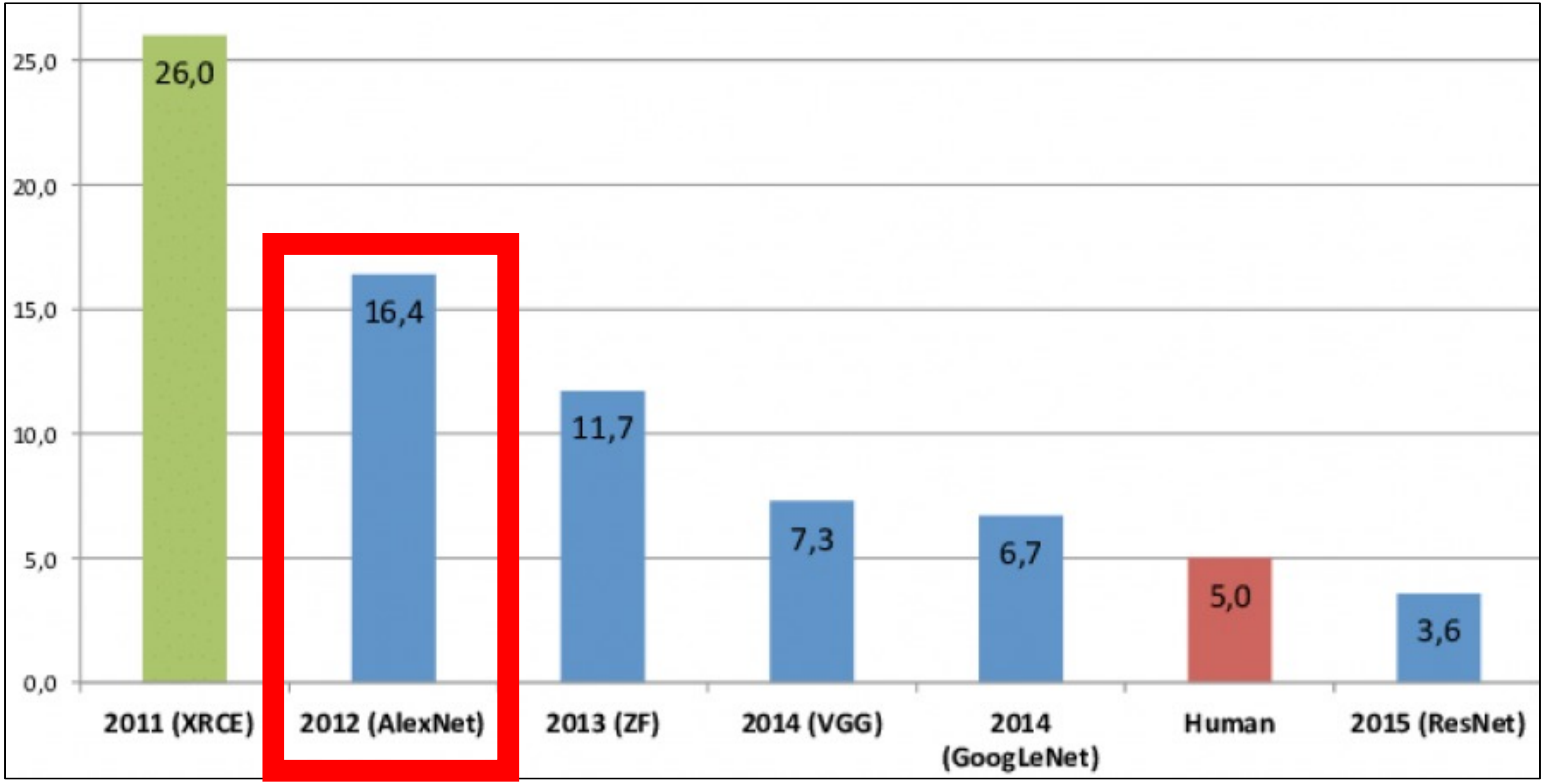


Figure Source: https://www.edge-ai-vision.com/2018/07/deep-learning-in-five-and-a-half-minutes/

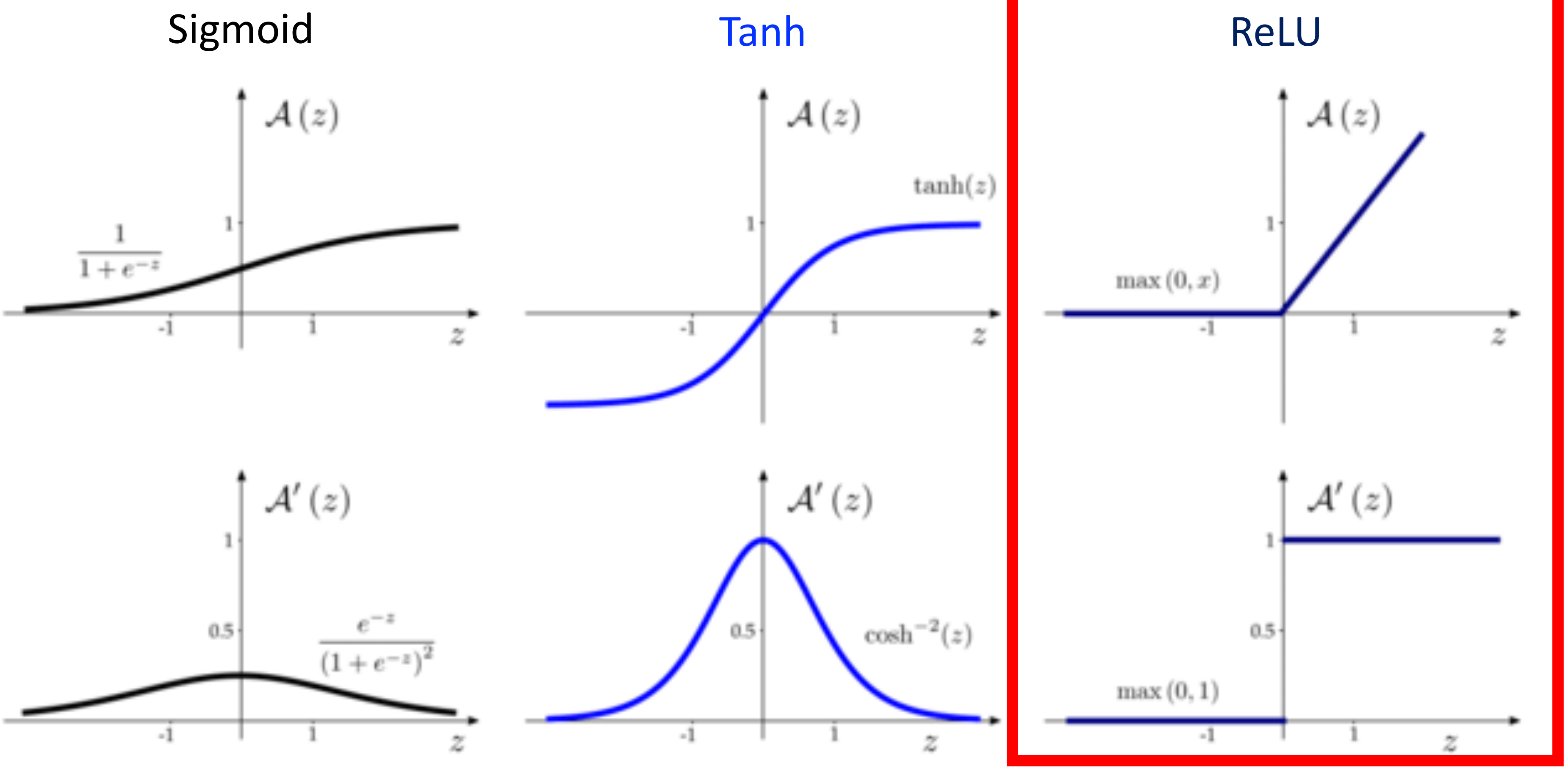
Why AlexNet?

Alex is the name of the paper's author ☺

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

Key Idea: Non-Saturating Activation Functions

Use activation functions with derivative value equal to 1 (i.e., 1x1x1… doesn't vanish)



Benefits:

- Fast to compute
- Can preserve gradient and so support efficient learning

Masi et al. Journal of the Mechanics and Physics of Solids. 2021

Architecture

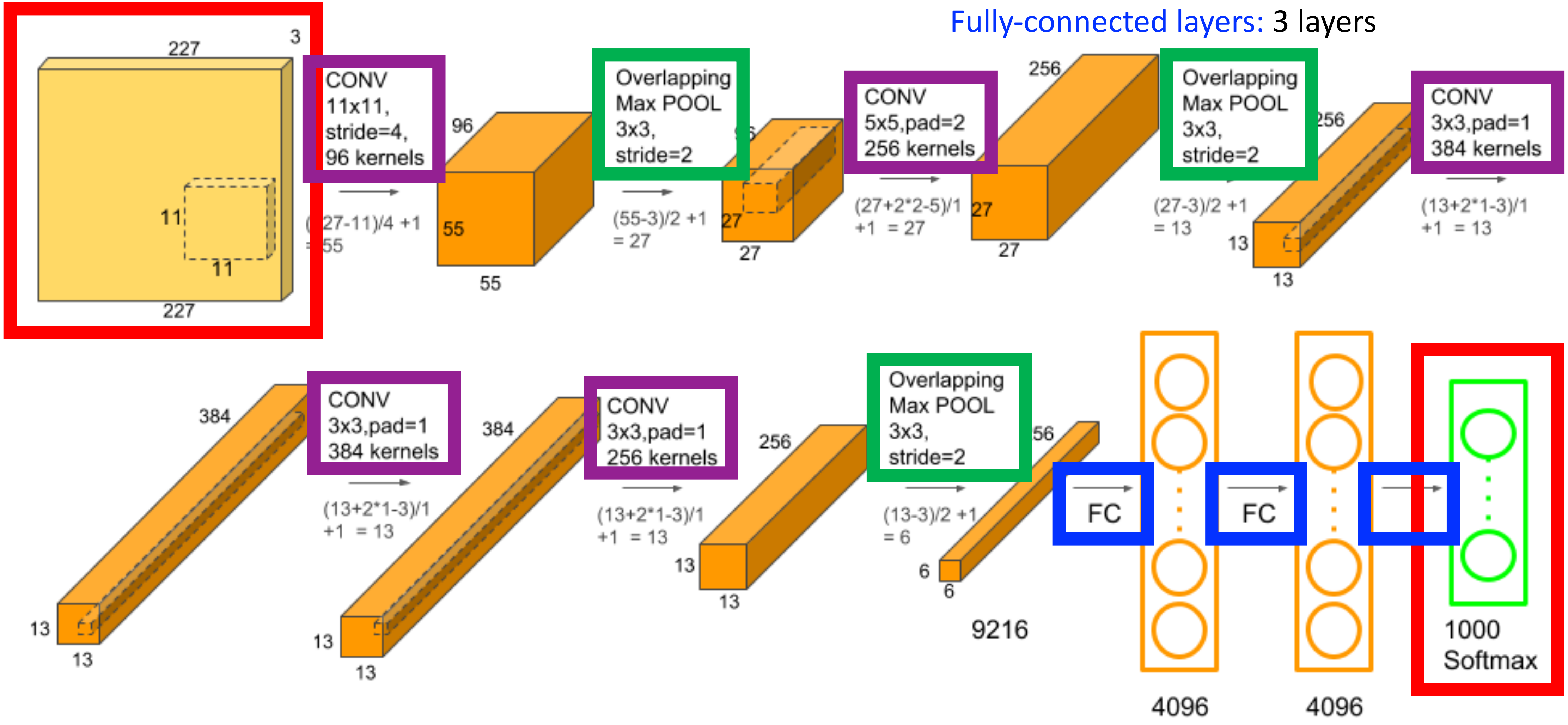
Input: RGB image resized to fixed input size

Output: 1000 class probabilities (sums to 1)

Convolutional layers: 5 layers

Pooling Layers: 3 layers

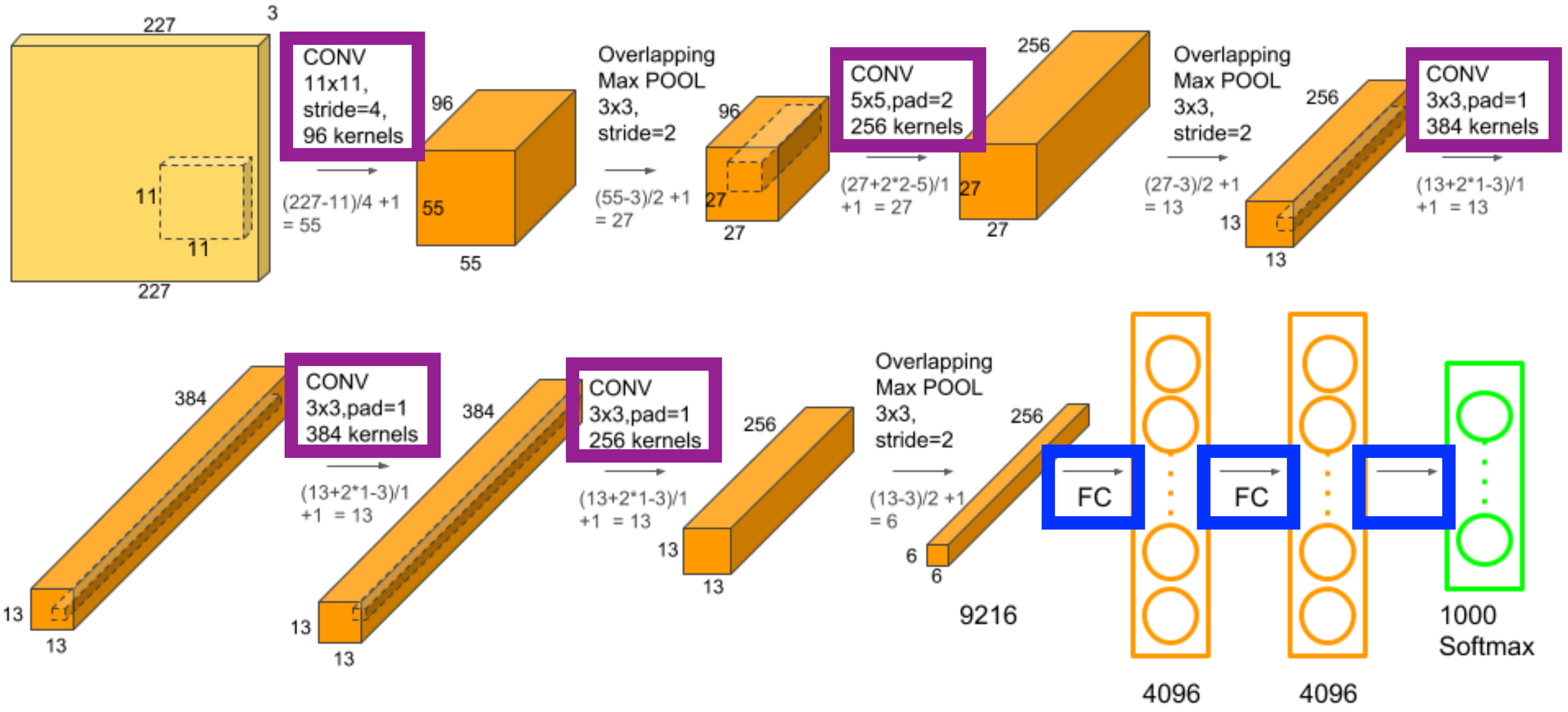
Fully-connected layers: 3 layers



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

Architecture

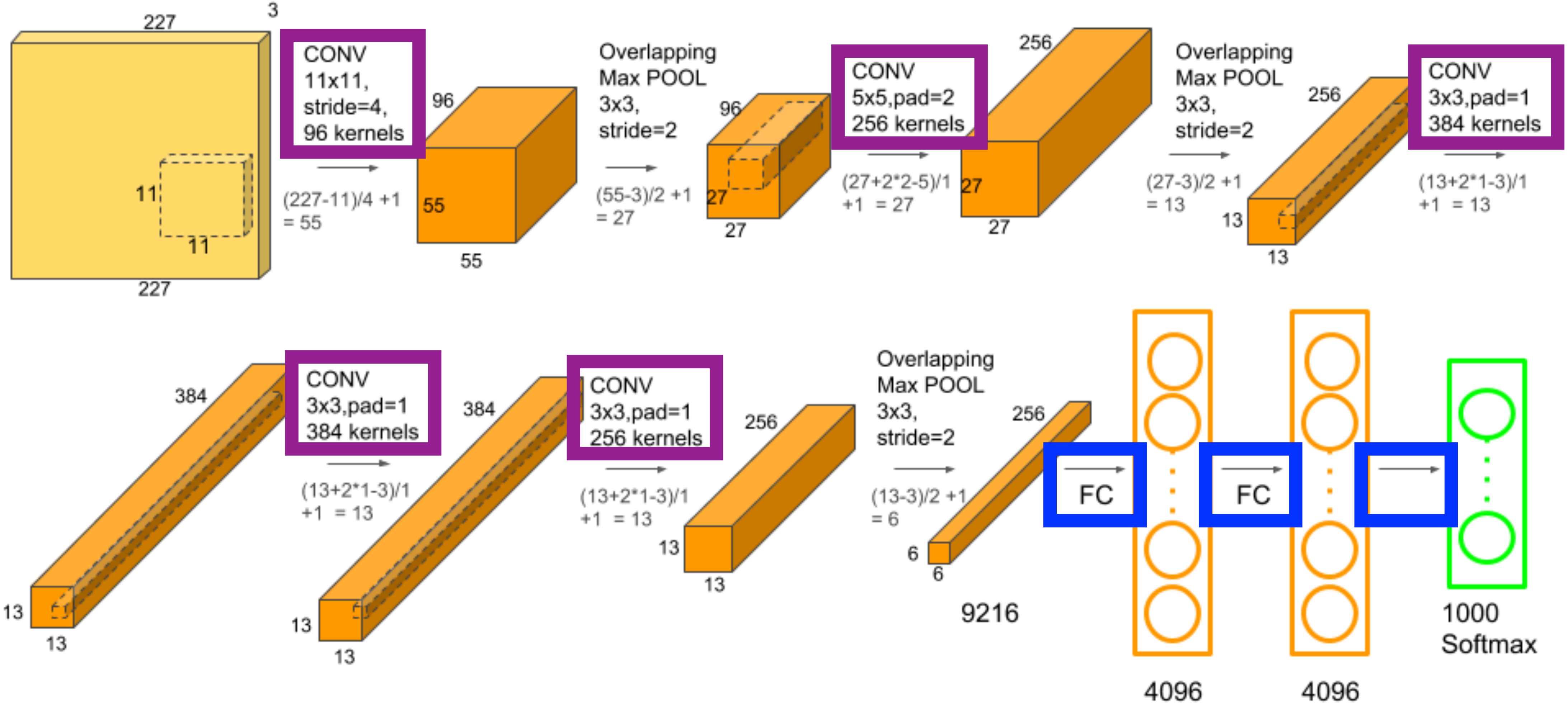
How many layers have model parameters that need to be learned?



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

Architecture

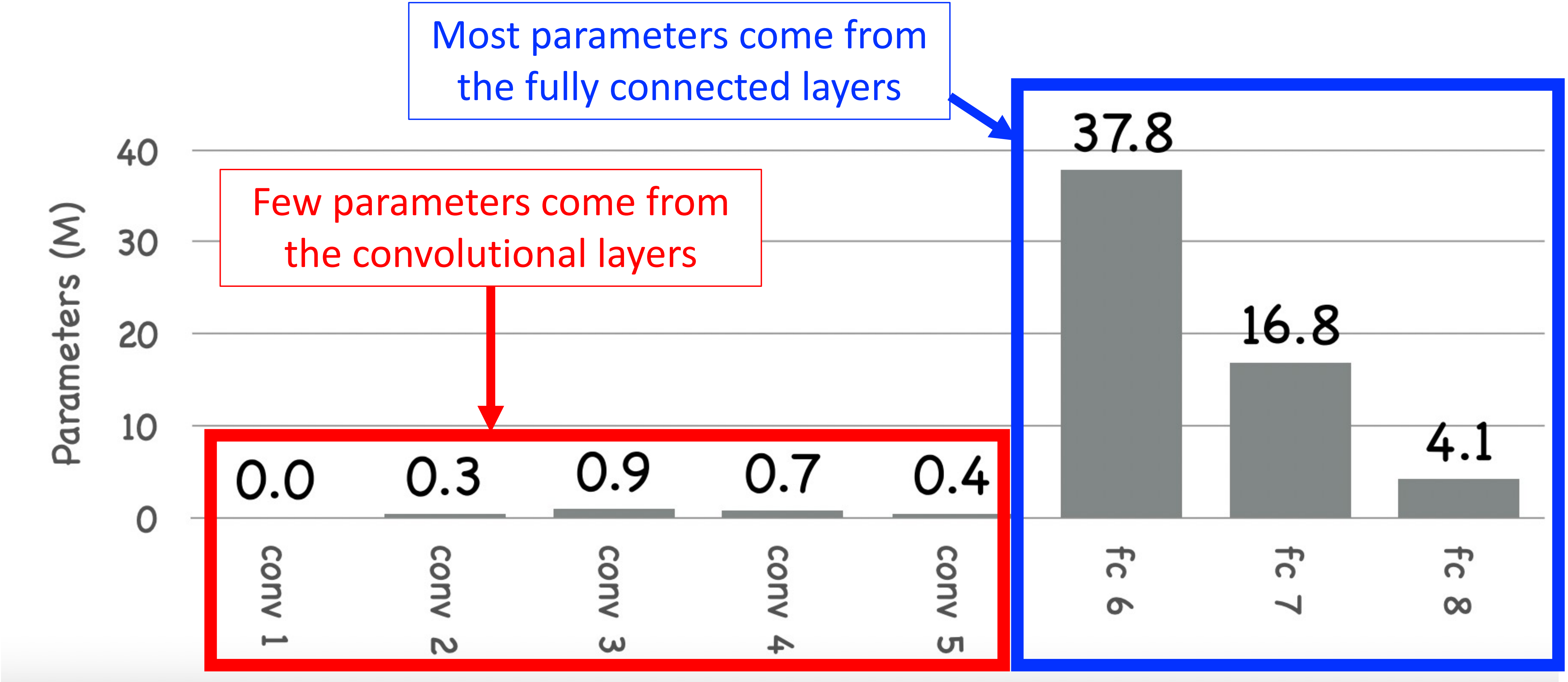
Altogether, 60 million model parameters must be learned!



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

Architecture

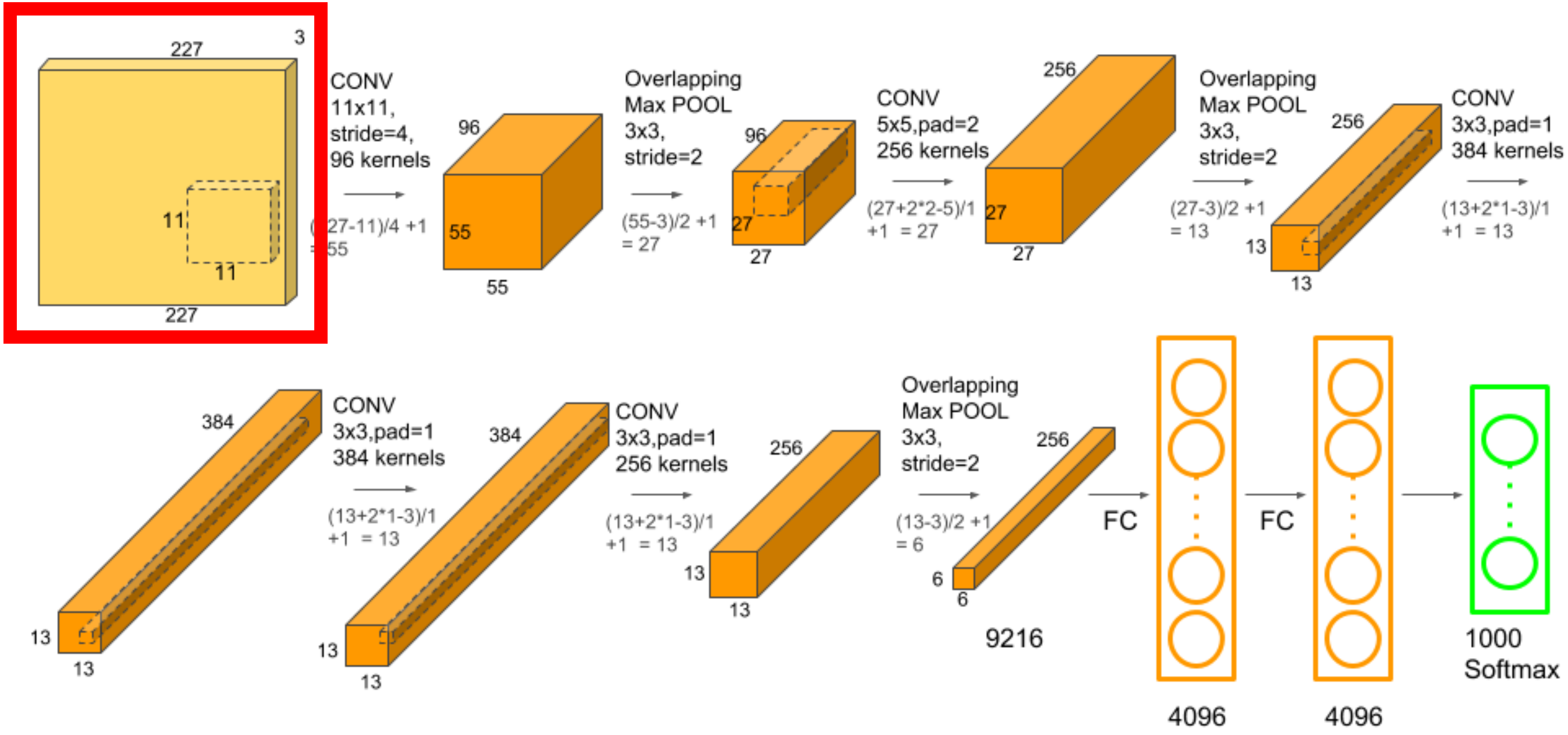
Altogether, 60 million model parameters must be learned!



Source: http://www.philkr.net/cs342/lectures/computer_vision/02.pdf

Input: RGB image resized to fixed input size

Architecture



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

Input Preprocessing

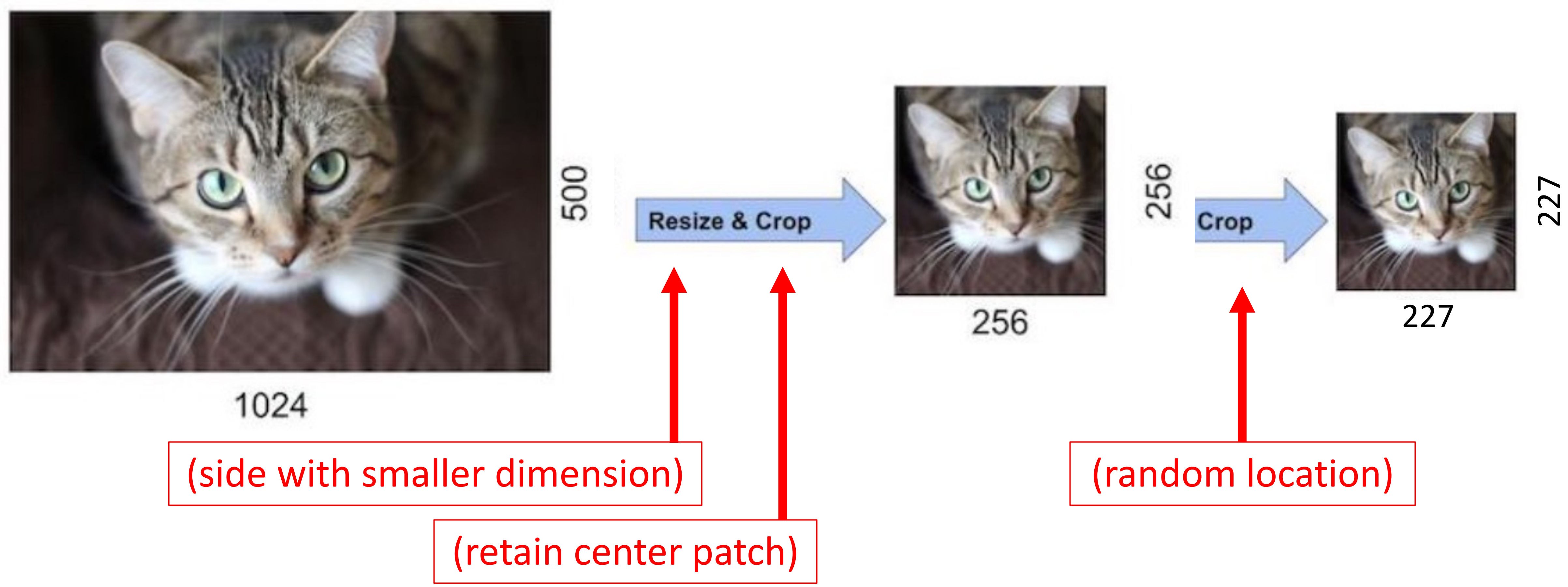
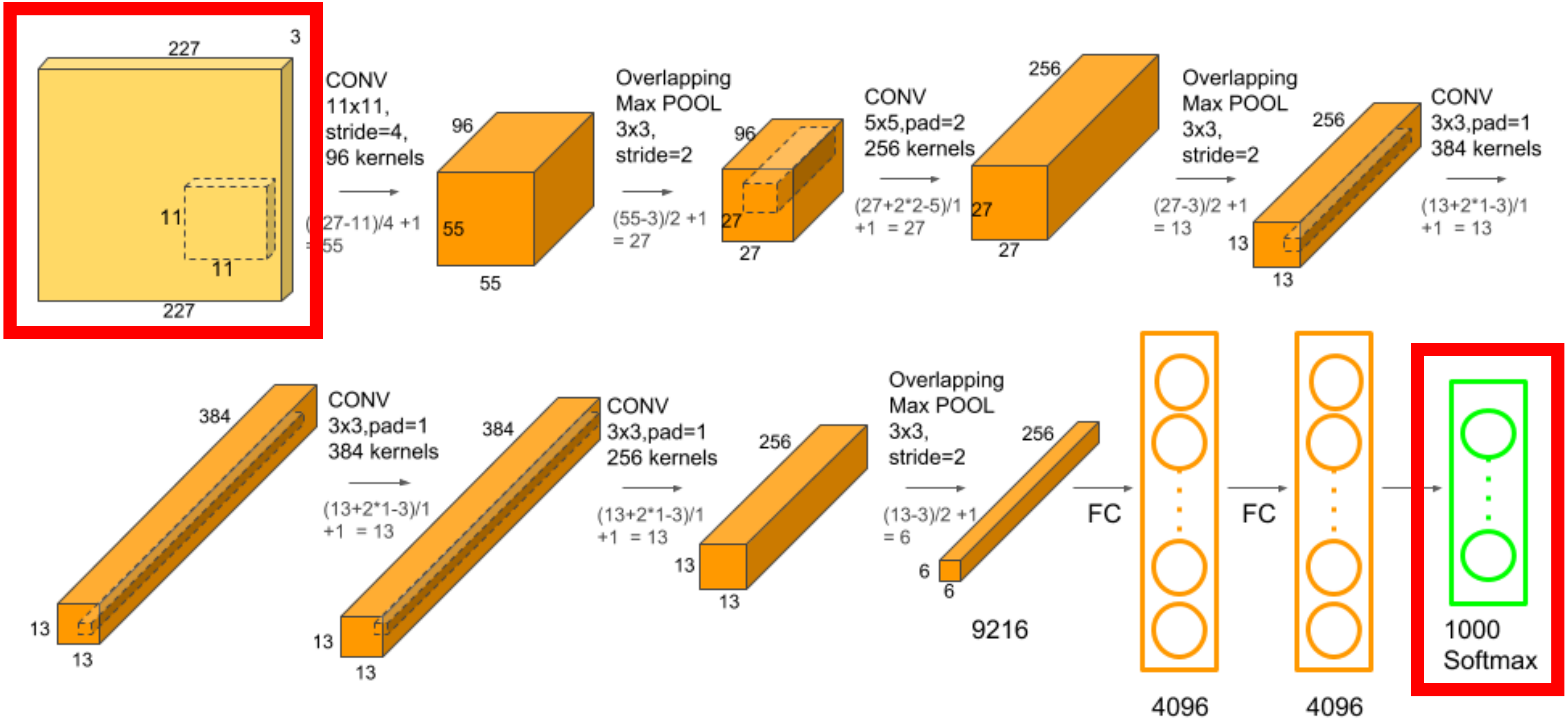


Image Source: https://learnopencv.com/understanding-alexnet/

Input: RGB image resized to fixed input size

Output: 1000 class probabilities (sums to 1)

Architecture



Source: https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

Architecture: Output Softmax Layer (Multiclass Classification)

Softmax: converts vector of scores into a probability distribution that sums to 1; e.g.,

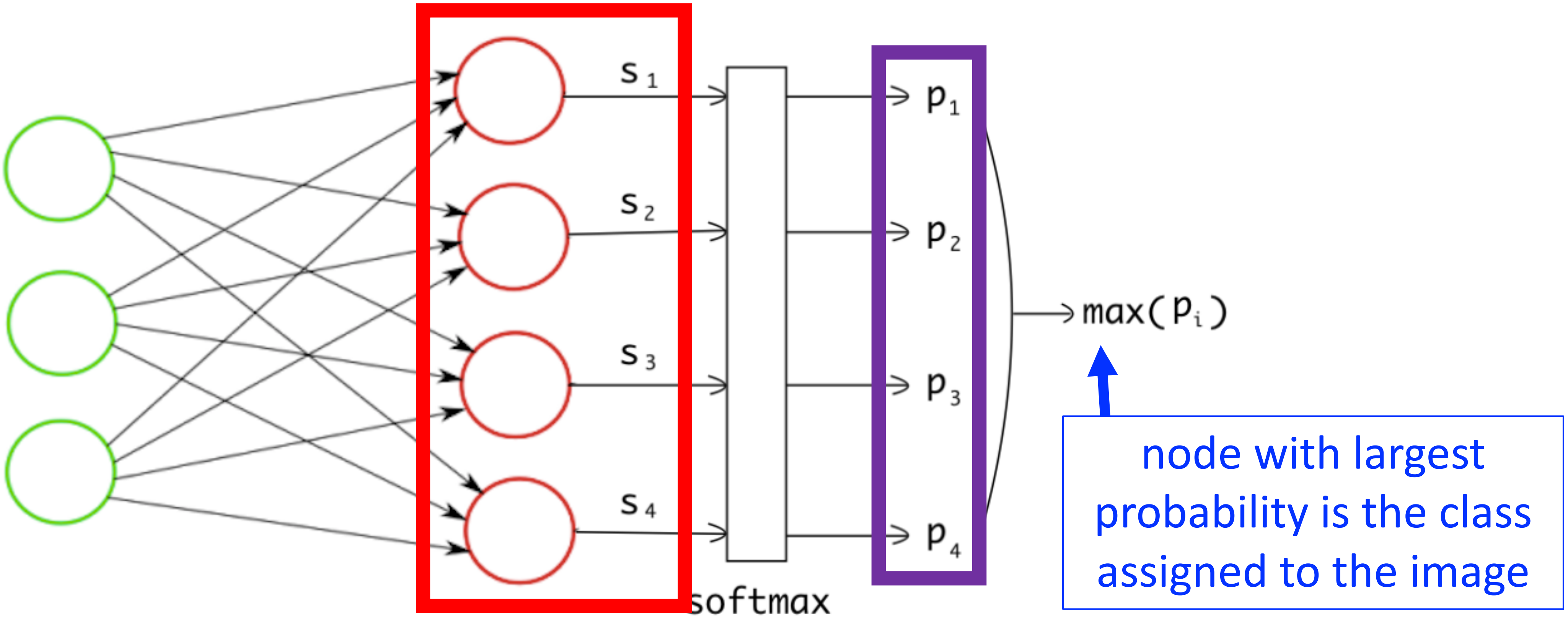
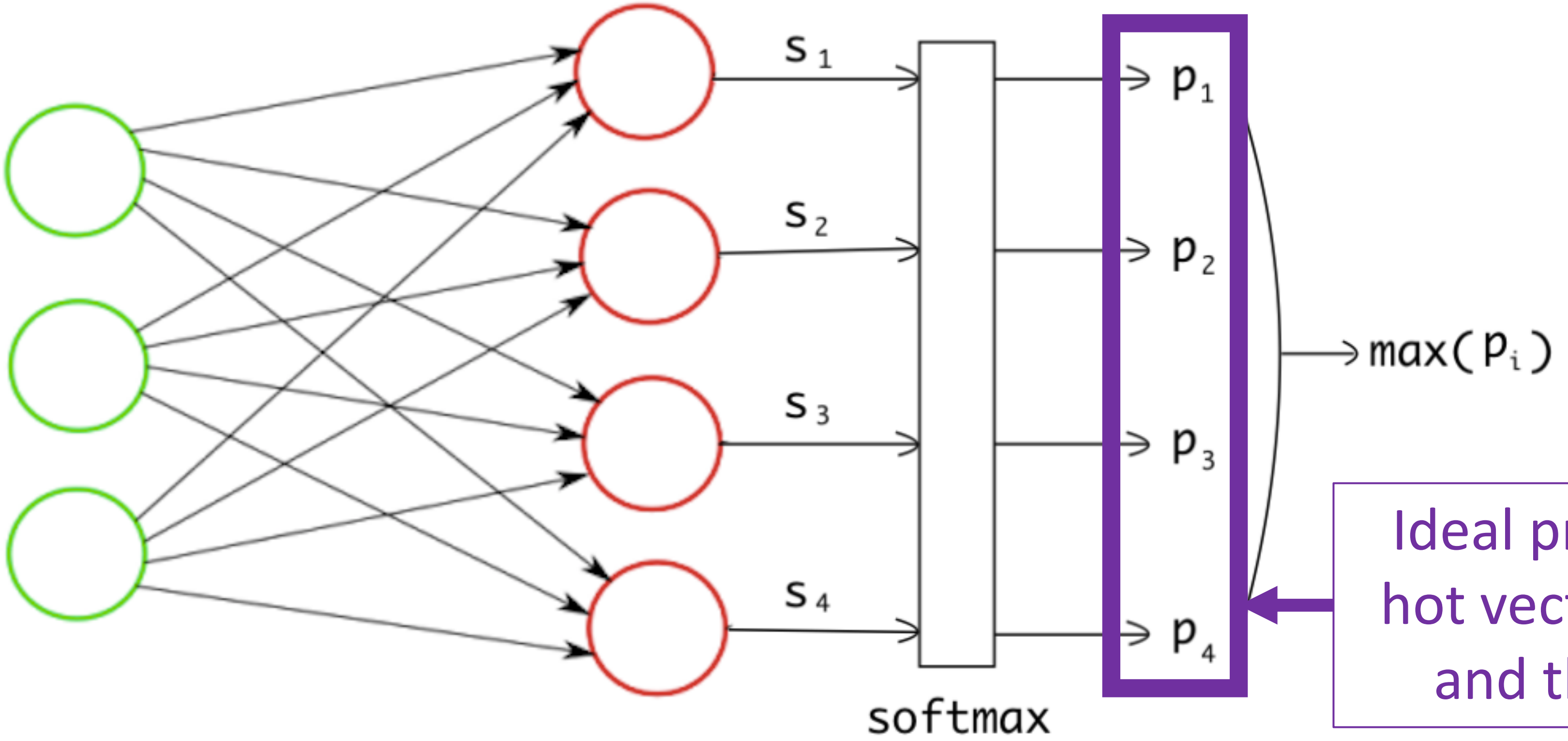


Figure Source: https://towardsdatascience.com/multi-label-image-classification-with-neural-network-keras-ddc1ab1afede

Architecture: Output Softmax Layer (Multiclass Classification)

Softmax: converts vector of scores into a probability distribution that sums to 1; e.g.,



Ideal prediction; 1 hot vector of one 1 and the rest 0s

Figure Source: https://towardsdatascience.com/multi-label-image-classification-with-neural-network-keras-ddc1ab1afede

Architecture: Output Softmax Layer (Multiclass Classification)

Softmax: converts vector of scores into a probability distribution that sums to 1

Remove negative values while preserving original order of scores using *e* (negative scores become slightly larger than 0 and positive values grow exponentially; choosing *e* rather than another exponent base simplifies subsequent math)

$$\sigma(\boxed{\mathbf{z}})_i = \frac{\boxed{e}^{\boxed{z_i}}}{\boxed{\sum_{j=1}^{K} \boxed{e}^{z_j}}}$$

i = 1, …, K

Number of classes (i.e., 1000)

Want to divide each node's score by sum of all entries to make them sum to 1 (normalization)

Useful tutorial: https://towardsdatascience.com/exploring-the-softmax-function-578c8b0fb15

Architecture: Output Softmax Layer (Multiclass Classification)

Softmax: converts vector of scores into a probability distribution that sums to 1; e.g.,

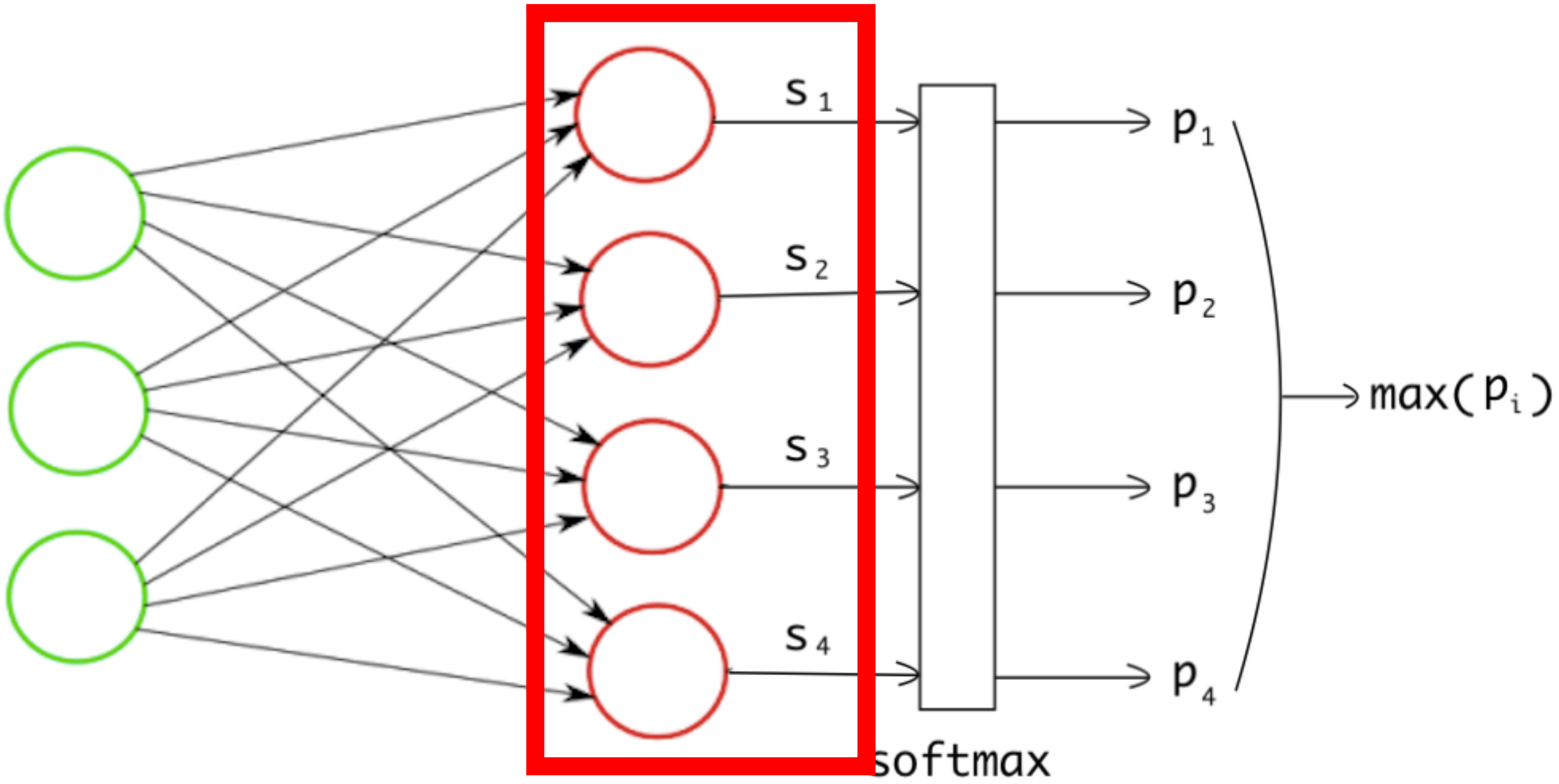


Figure Source: https://towardsdatascience.com/multi-label-image-classification-with-neural-network-keras-ddc1ab1afede

Architecture: Output Softmax Layer (Multiclass Classification)

Softmax: converts vector of scores into a probability distribution that sums to 1; e.g.,

	Scoring Function
Dog	-3.44
Cat	1.16
Boat	-0.81
Airplane	3.91

Architecture: Output Softmax Layer (Multiclass Classification)

Softmax: converts vector of scores into a probability distribution that sums to 1; e.g.,

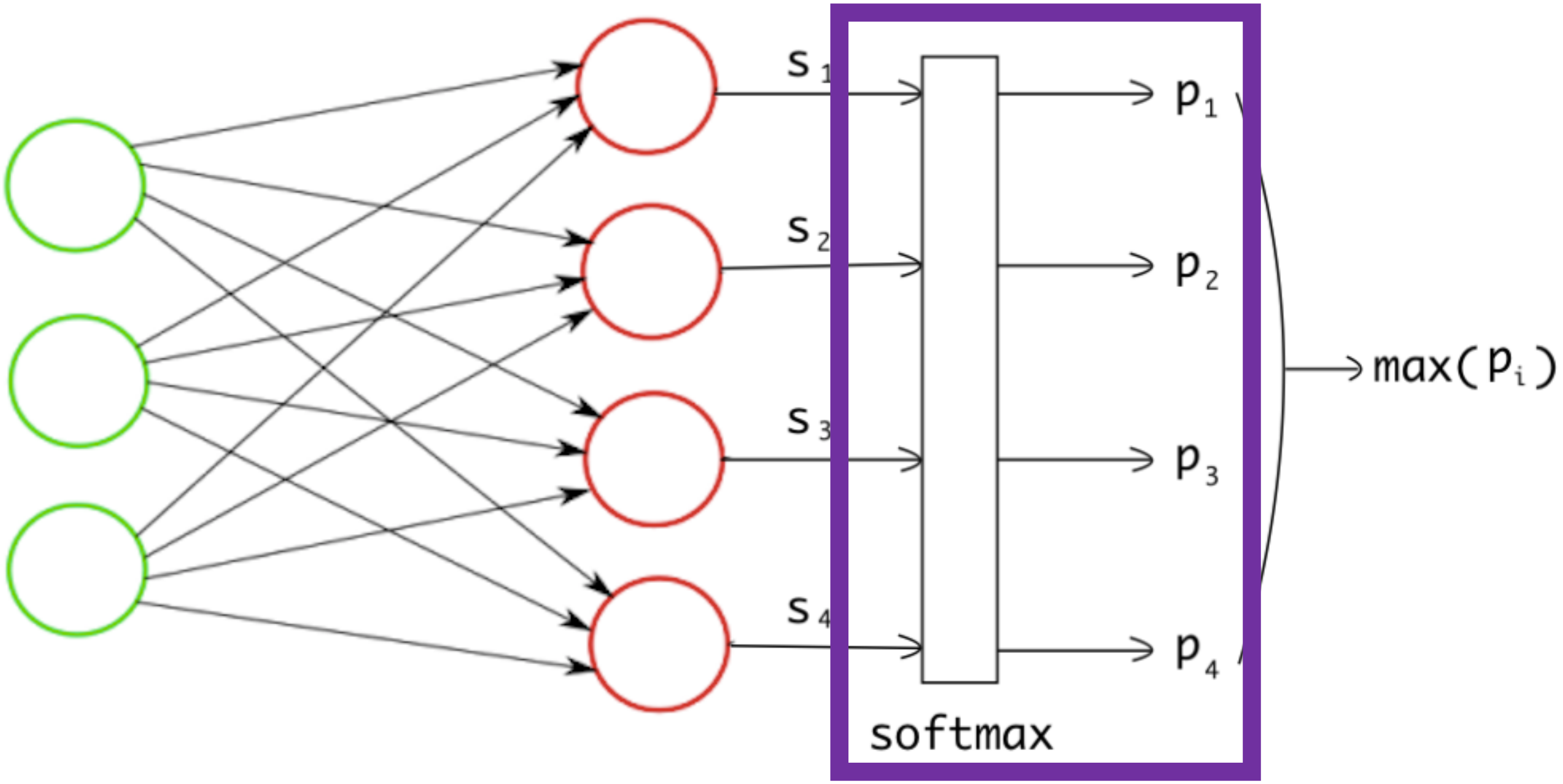


Figure Source: https://towardsdatascience.com/multi-label-image-classification-with-neural-network-keras-ddc1ab1afede

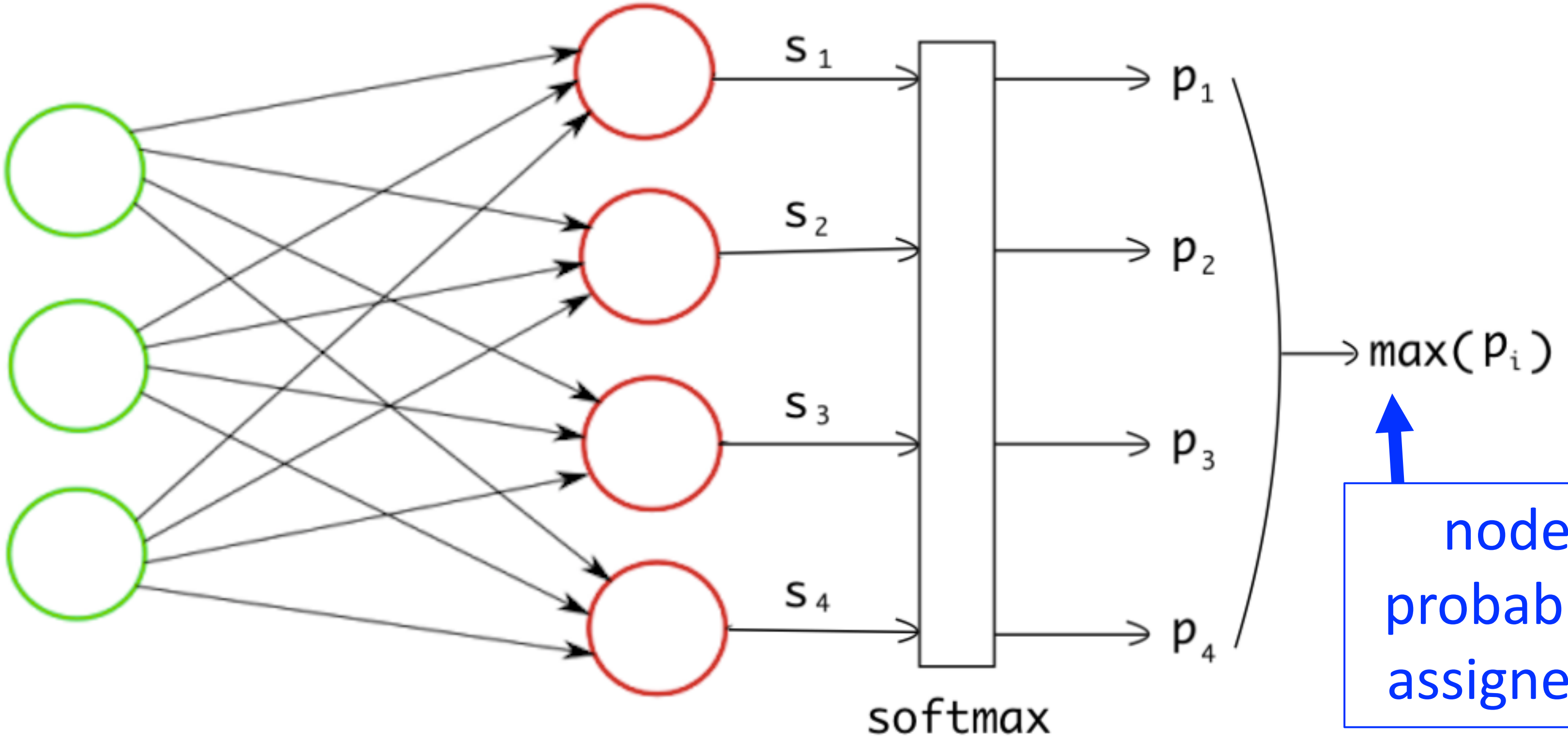
Architecture: Output Softmax Layer (Multiclass Classification)

Softmax: converts vector of scores into a probability distribution that sums to 1; e.g.,

	Scoring Function	Unnormalized Probabilities e^{z_i}	Normalized Probabilities $\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$
Dog	-3.44	0.0321	0.0006
Cat	1.16	3.1899	0.0596
Boat	-0.81	0.4449	0.0083
Airplane	3.91	49.8990	0.9315

Architecture: Output Softmax Layer (Multiclass Classification)

Softmax: converts vector of scores into a probability distribution that sums to 1; e.g.,



node with largest probability is the class assigned to the image

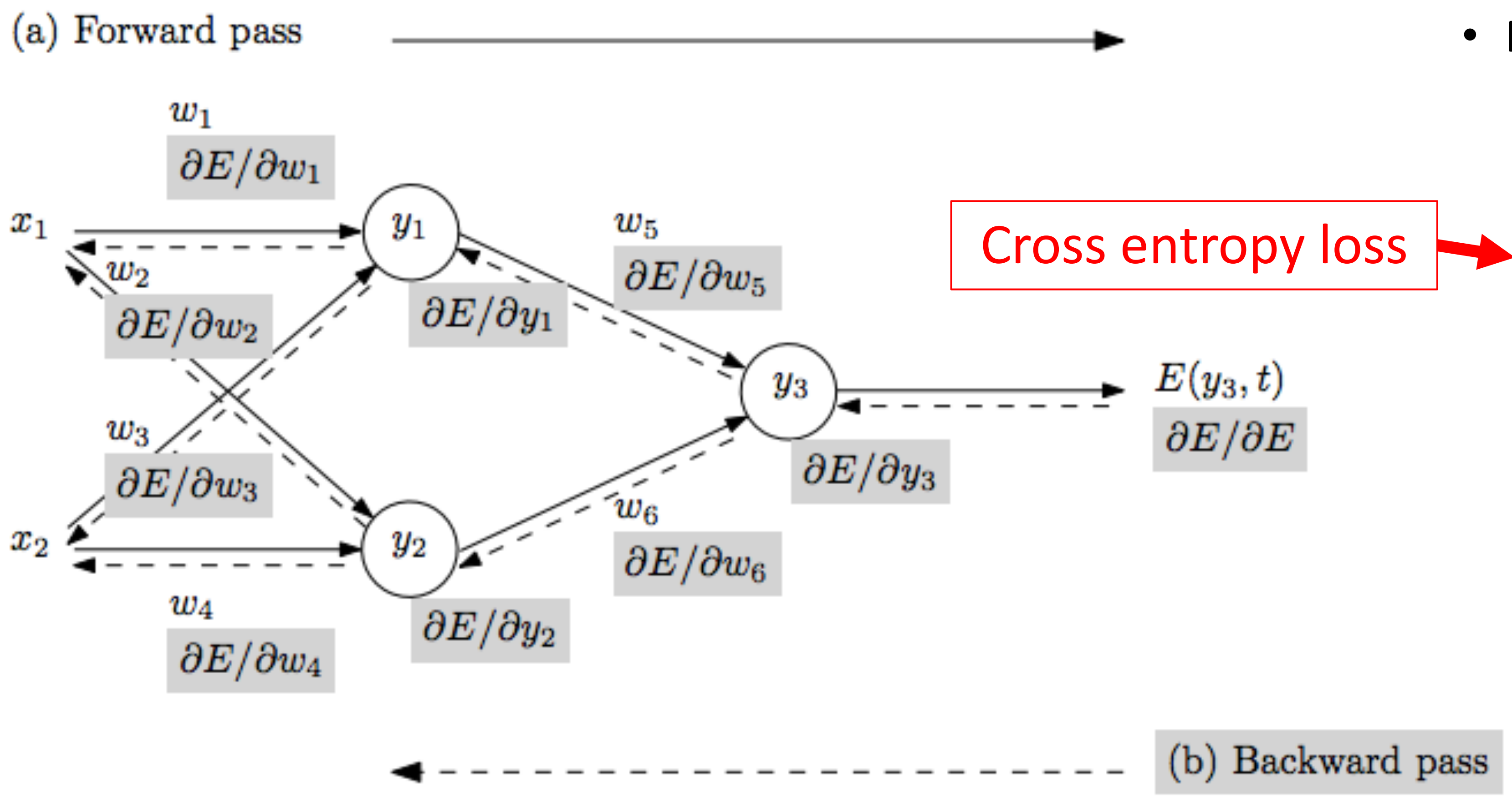
Figure Source: https://towardsdatascience.com/multi-label-image-classification-with-neural-network-keras-ddc1ab1afede

Architecture: Output Softmax Layer (Multiclass Classification)

Softmax: converts vector of scores into a probability distribution that sums to 1; e.g.,

	Scoring Function	Unnormalized Probabilities	Normalized Probabilities
Dog	-3.44	0.0321	0.0006
Cat	1.16	3.1899	0.0596
Boat	-0.81	0.4449	0.0083
Airplane	3.91	49.8990	0.9315

Algorithm Training: Recall How NNs Learn



- Repeat until stopping criterion met:
  1. **Forward pass**: propagate training data through model to make predictions
  2. **Error quantification**: measure dissatisfaction with a model's predictions on training data
  3. **Backward pass**: using predicted output, calculate gradients backward to assign blame to each model parameter
  4. Update each parameter using calculated gradients

https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/

Algorithm Training: Measure Cross Entropy Loss

Measure distance between predicted and true class distribution for each example

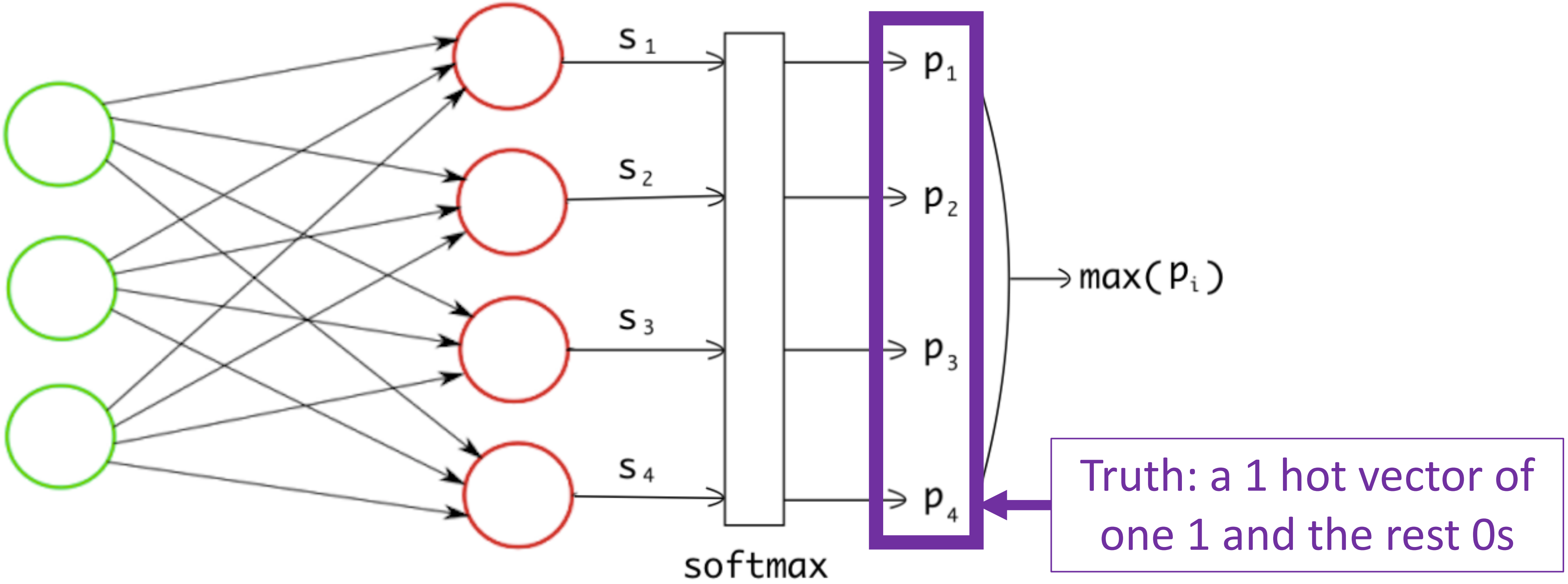


Figure Source: https://towardsdatascience.com/multi-label-image-classification-with-neural-network-keras-ddc1ab1afede

Algorithm Training: Measure Cross Entropy Loss

Probability distribution of true class

Probability distribution of predicted class

Number of classes?

Recall, truth is set to 1 for one class and 0 otherwise

Observed features

Simplifies to the log of the predicted probability for the correct class (i.e., negative log likelihood loss)

$$
\begin{aligned}
L_{\mathrm{CE}}(\hat{y}, y) &= -\sum_{k=1}^{K} y_k \log \hat{y}_k \\
&= -\sum_{k=1}^{K} y_k \log \hat{p}(y = k \mid x) \\
&= -\log \hat{y}_k, \quad \text{(where } k \text{ is the correct class)} \\
&= -\log \frac{\exp(w_k \cdot x + b_k)}{\sum_{j=1}^{K} \exp(w_j \cdot x + b_j)}
\end{aligned}
$$

Excellent background: https://web.stanford.edu/~jurafsky/slp3/5.pdf

Algorithm Training: Measure Cross Entropy Loss

Probability distribution of predicted class

Probability distribution of true class

Number of classes?

Recall, truth is set to 1 for one class and 0 otherwise

Observed features

$$
\begin{aligned}
L_{\mathrm{CE}}(\hat{y}, y) &= -\sum_{k=1}^{K} y_k \log \hat{y}_k \\
&= -\sum_{k=1}^{K} y_k \log \hat{p}(y=k|x) \\
&= -\log \hat{y}_k, \quad \text{(where } k \text{ is the correct class)} \\
&= -\log \frac{\exp(w_k \cdot x + b_k)}{\sum_{j=1}^{K} \exp(w_j \cdot x + b_j)}
\end{aligned}
$$

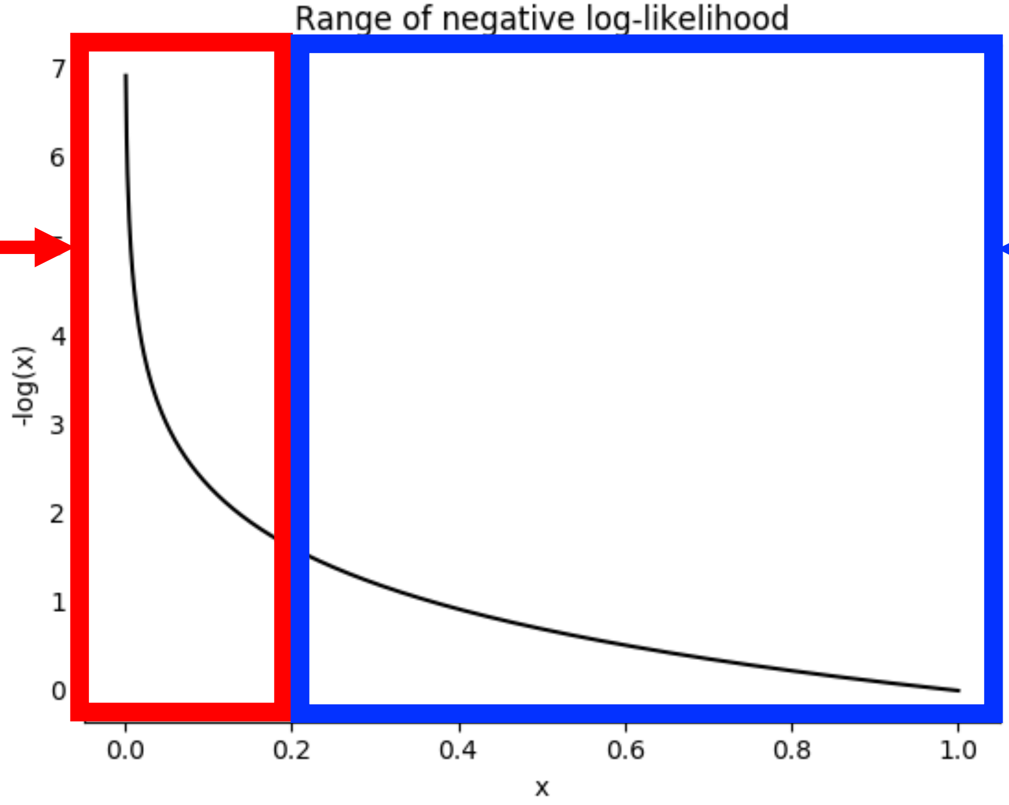
What is the range of possible values?

- Minimum: 0 (negative log of 1)
- Maximum: Infinity (negative log of 0)

Excellent background: https://web.stanford.edu/~jurafsky/slp3/5.pdf

Algorithm Training: Measure Cross Entropy Loss

Greater penalty when predicted probability of true class is confidently wrong



Lesser penalty otherwise

What is the range of possible values?

- Minimum: 0 (negative log of 1)
- Maximum: Infinity (negative log of 0)

$$= -\log \frac{\exp(w_k \cdot x + b_k)}{\sum_{j=1}^{K} \exp(w_j \cdot x + b_j)}$$

Figure source: https://ljvmiranda921.github.io/notebook/2017/08/13/softmax-and-the-negative-log-likelihood/

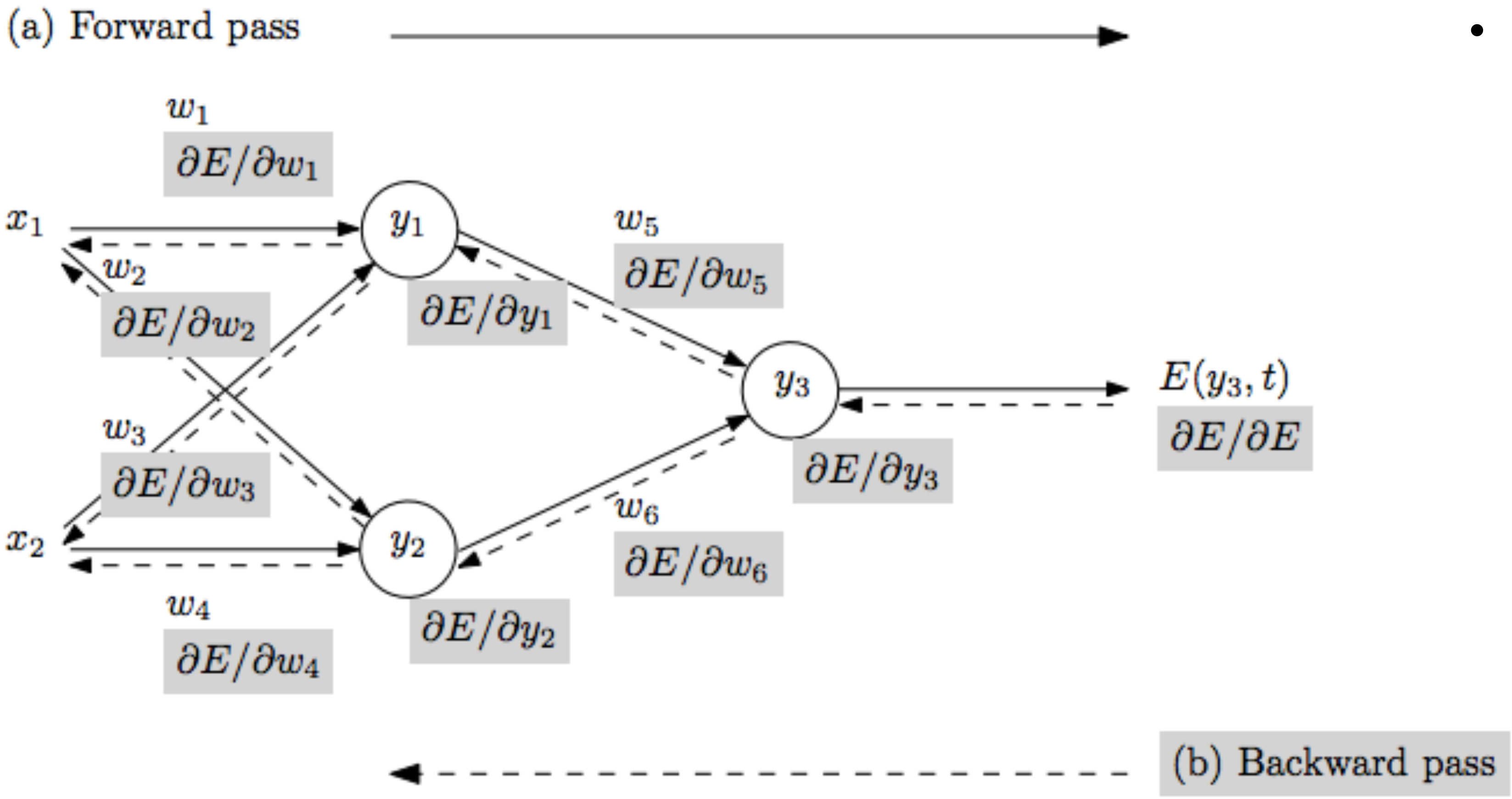
Algorithm Training: Measure Cross Entropy Loss

e.g., What would be the loss for this example if the true class label is cat?

$$= -\log \hat{y}_k, \quad \text{(where } k \text{ is the correct class)}$$
$$= -\log (0.0596) = 2.82$$

	Scoring Function	Unnormalized Probabilities	Normalized Probabilities
Dog	-3.44	0.0321	0.0006
Cat	1.16	3.1899	0.0596
Boat	-0.81	0.4449	0.0083
Airplane	3.91	49.8990	0.9315

Algorithm Training

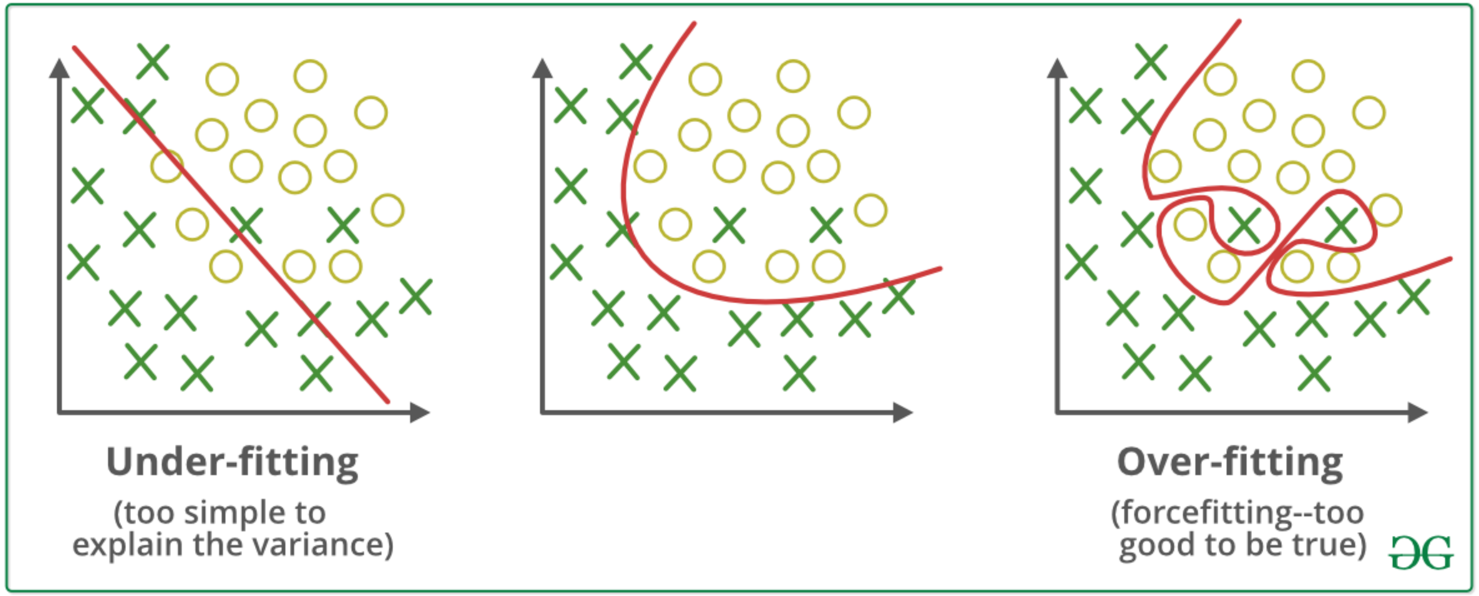


- Repeat until stopping criterion met:
  1. **Forward pass**: propagate training data through model to make predictions
  2. **Error quantification**: measure dissatisfaction with a model's predictions on training data
  3. **Backward pass**: using predicted output, calculate gradients backward to assign blame to each model parameter
  4. Update each parameter using calculated gradients

https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/

Algorithm Training: Challenge Is Overfitting

- Overfitting is risk for models with larger representational capacity (i.e., # of parameters); AlexNet has 60 million parameters!



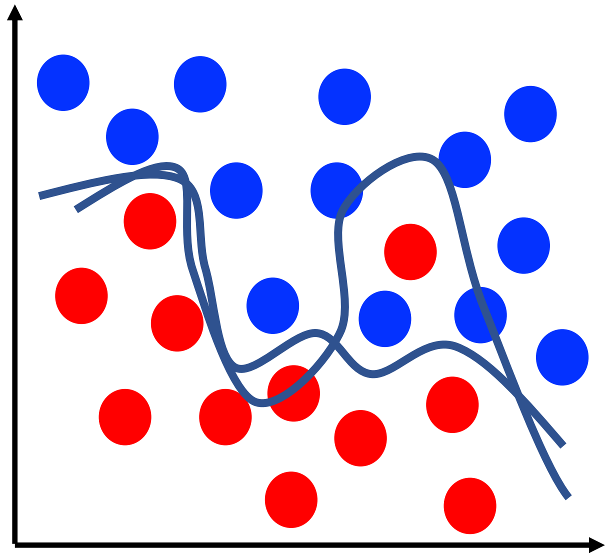
- Model learns to model **noise!**

Figure source: https://towardsdatascience.com/underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6fe4a8a49dbf

AlexNet Remedies for Overfitting

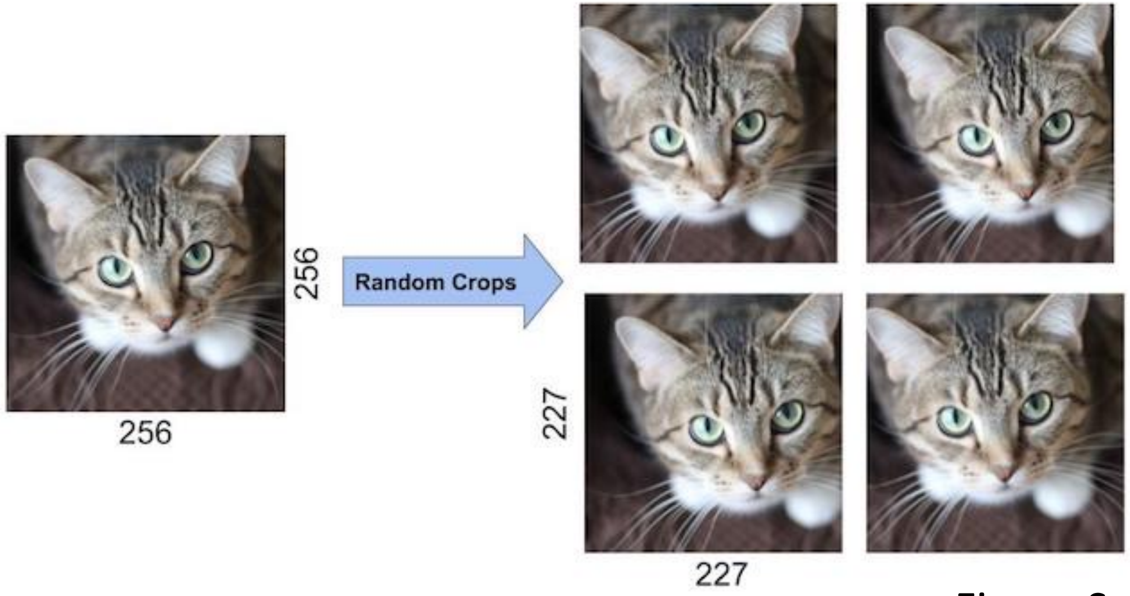
- Overfitting is risk for models with larger representational capacity (i.e., # of parameters); AlexNet has 60 million parameters!
    1. Data augmentation: add more training data; e.g., intuitively,

Adding training data



AlexNet Remedies for Overfitting

- Overfitting is risk for models with larger representational capacity (i.e., # of parameters); AlexNet has 60 million parameters!
  1. Data augmentation
     1. Random patches and their mirror images (2048x more data)
     2. Adjust RGB channels (using PCA to add multiples of principal components)



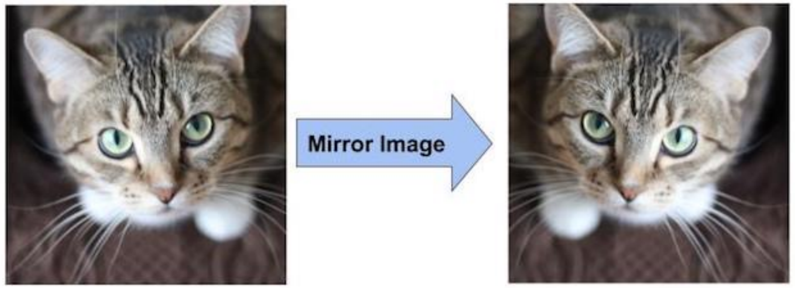
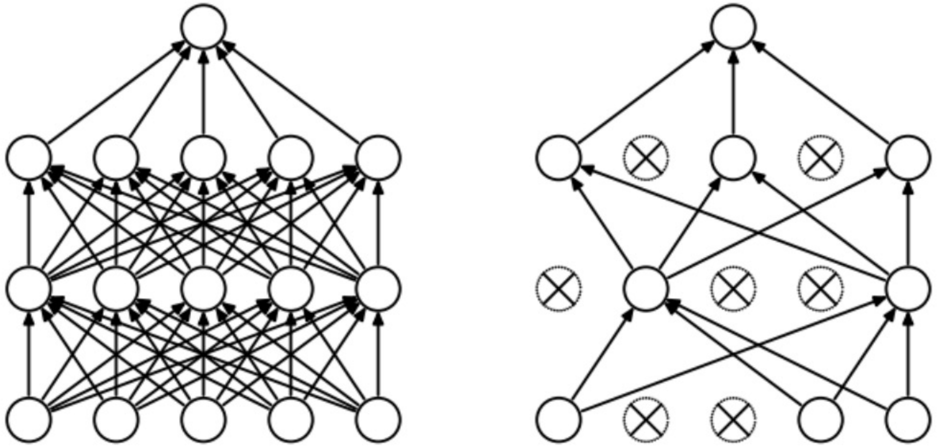


Figure Source: https://learnopencv.com/understanding-alexnet/

AlexNet Remedies for Overfitting

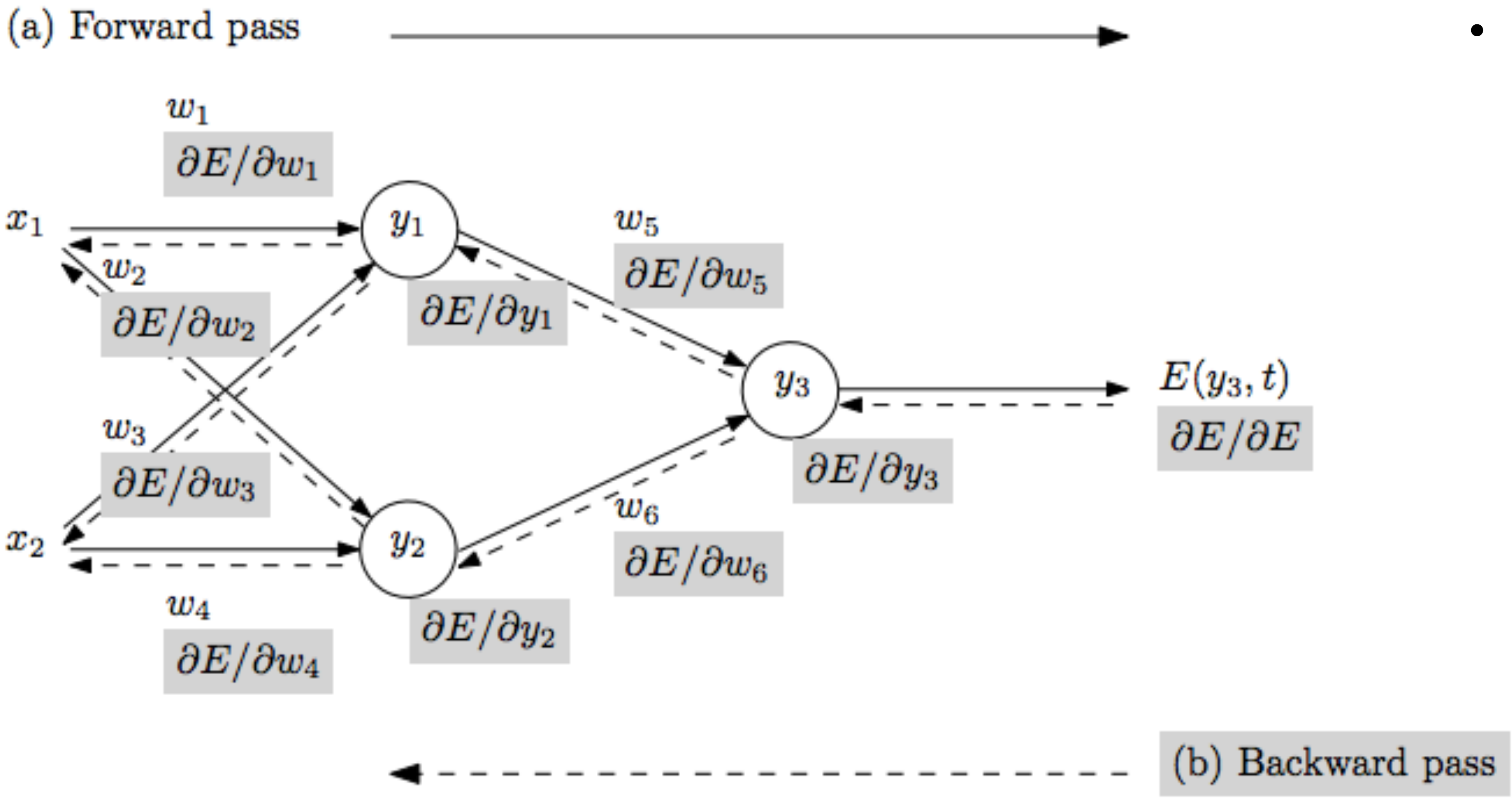
- Overfitting is risk for models with larger representational capacity (i.e., # of parameters); AlexNet has 60 million parameters!
  1. Data augmentation
     1. Random patches and their mirror images (2048x more data)
     2. Adjust RGB channels (using PCA to add multiples of principal components)
  2. Dropout (50% of nodes for first two fully connected layers); mimics ensembles by learning to solve same problem with different subnetworks



(a) Standard Neural Net (b) After applying dropout.

Figure Source: Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting." JMLR, 2014.

Algorithm Training: 90 Epochs on ImageNet



- Repeat until stopping criterion met:
  1. **Forward pass**: propagate training data through model to make predictions
  2. **Error quantification**: measure dissatisfaction with a model's predictions on training data
  3. **Backward pass**: using predicted output, calculate gradients backward to assign blame to each model parameter
  4. Update each parameter using calculated gradients

https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/

AlexNet Analysis

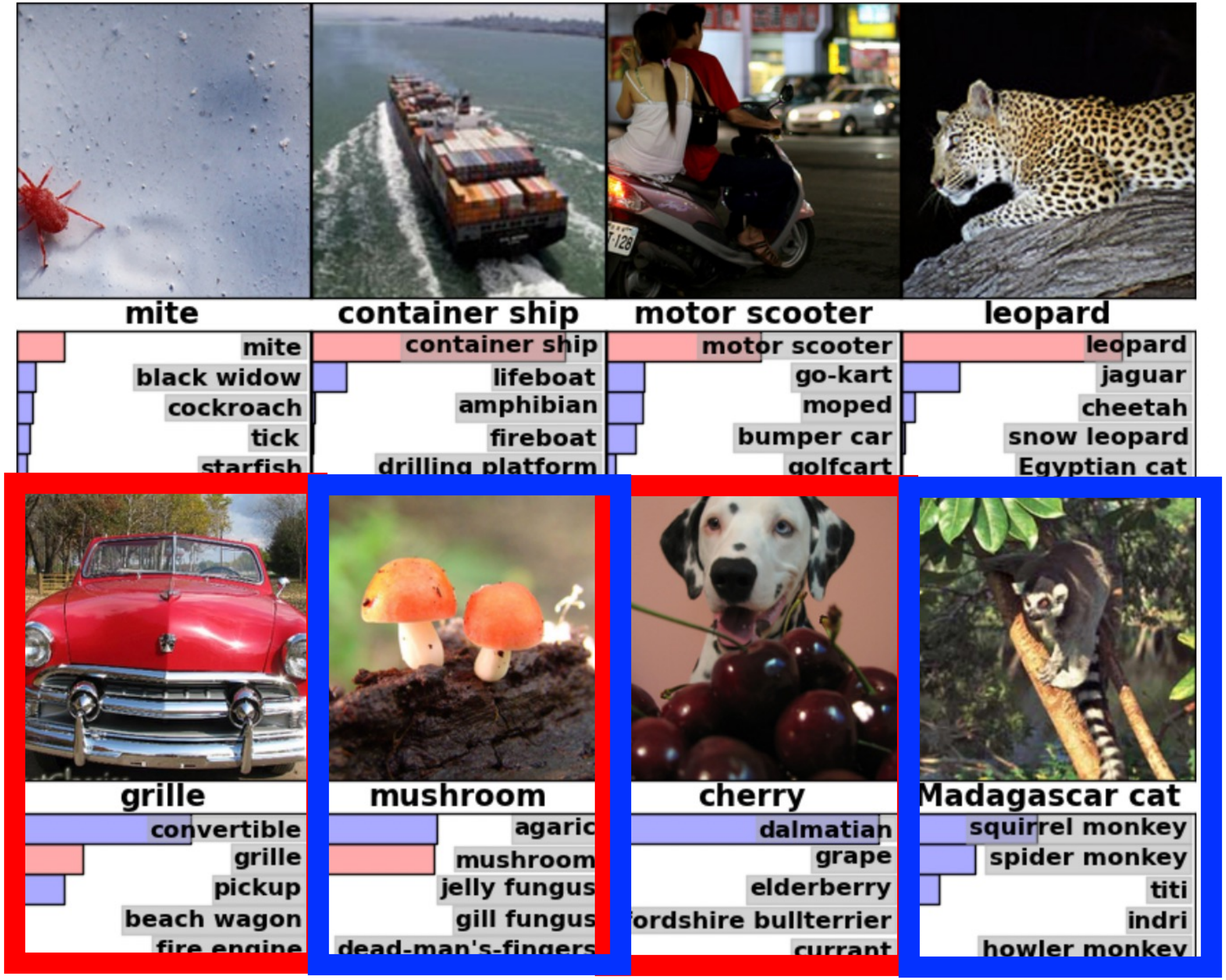
8 examples of predictions, correct and incorrect

When/why might the model succeed?

- Single well-defined object (even if off-centered)

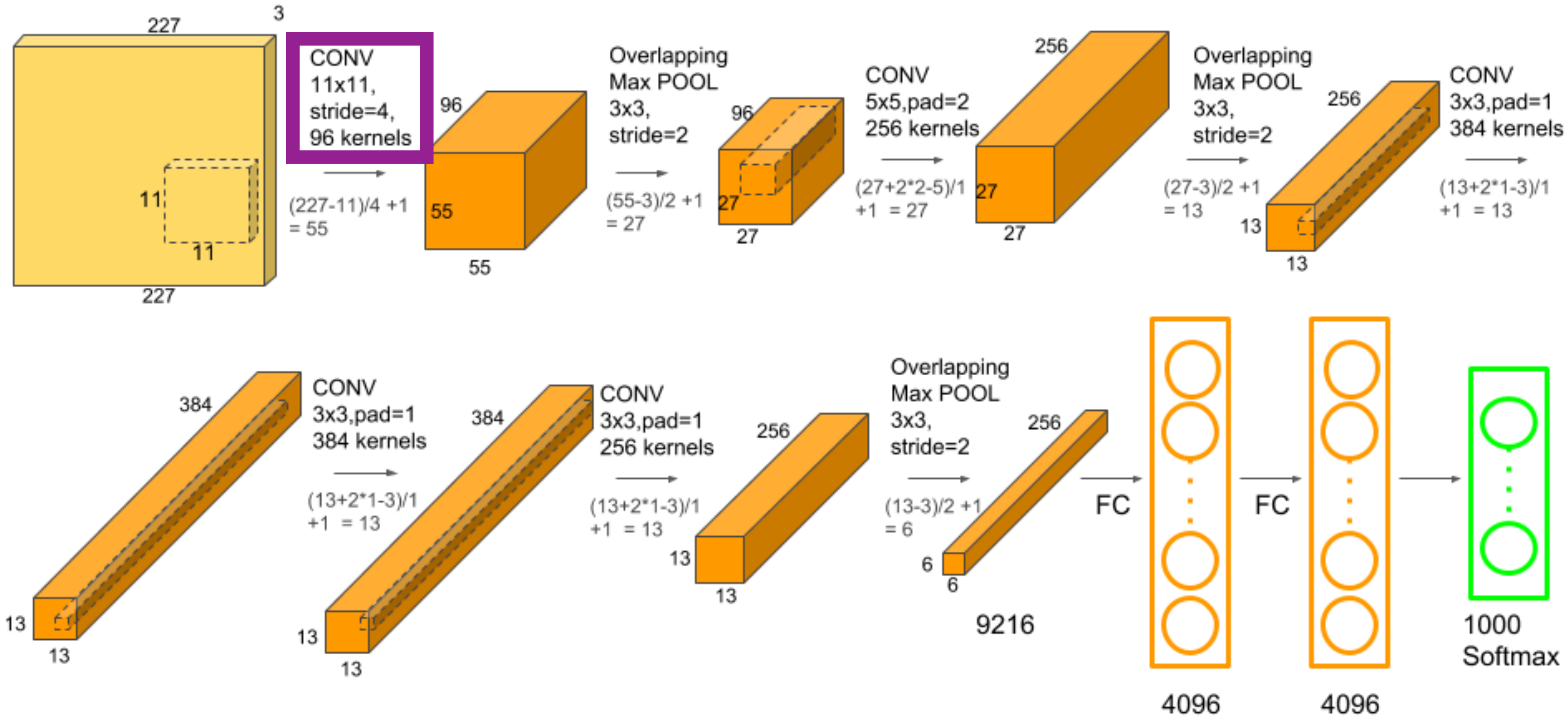
When/why might the model fail?

- Ambiguity
- Similar categories

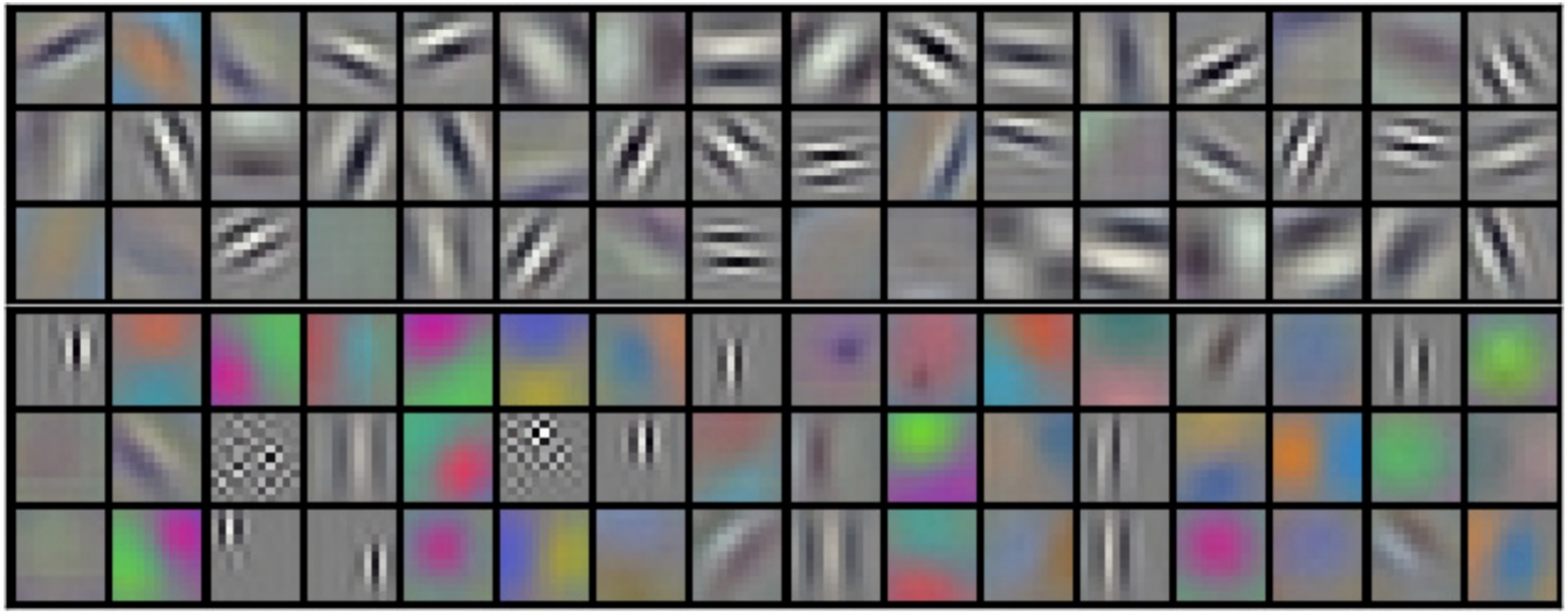


Krizhevsky et al. ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM, 2017

AlexNet: Inspecting What It Learned



AlexNet: Inspecting What It Learned (96 Filters)



Model learned filters that select based on frequency, orientation, and color!
(Aligns with Hubel & Weisel’s findings for how vision systems work)

AlexNet: Key Tricks for Going Deeper

- ReLU instead of sigmoid or tanh activation functions

- Regularization techniques; e.g.,
    1. Data augmentation
    2. Dropout in fully connected layers

- Trained across two GPUs

Object Recognition: Today's Topics

- ImageNet Challenge Top Performers
- Baseline Model: AlexNet
- VGG
- ResNet
- Discussion

VGG: A Deeper CNN

Progress of models on ImageNet (Top 5 Error)

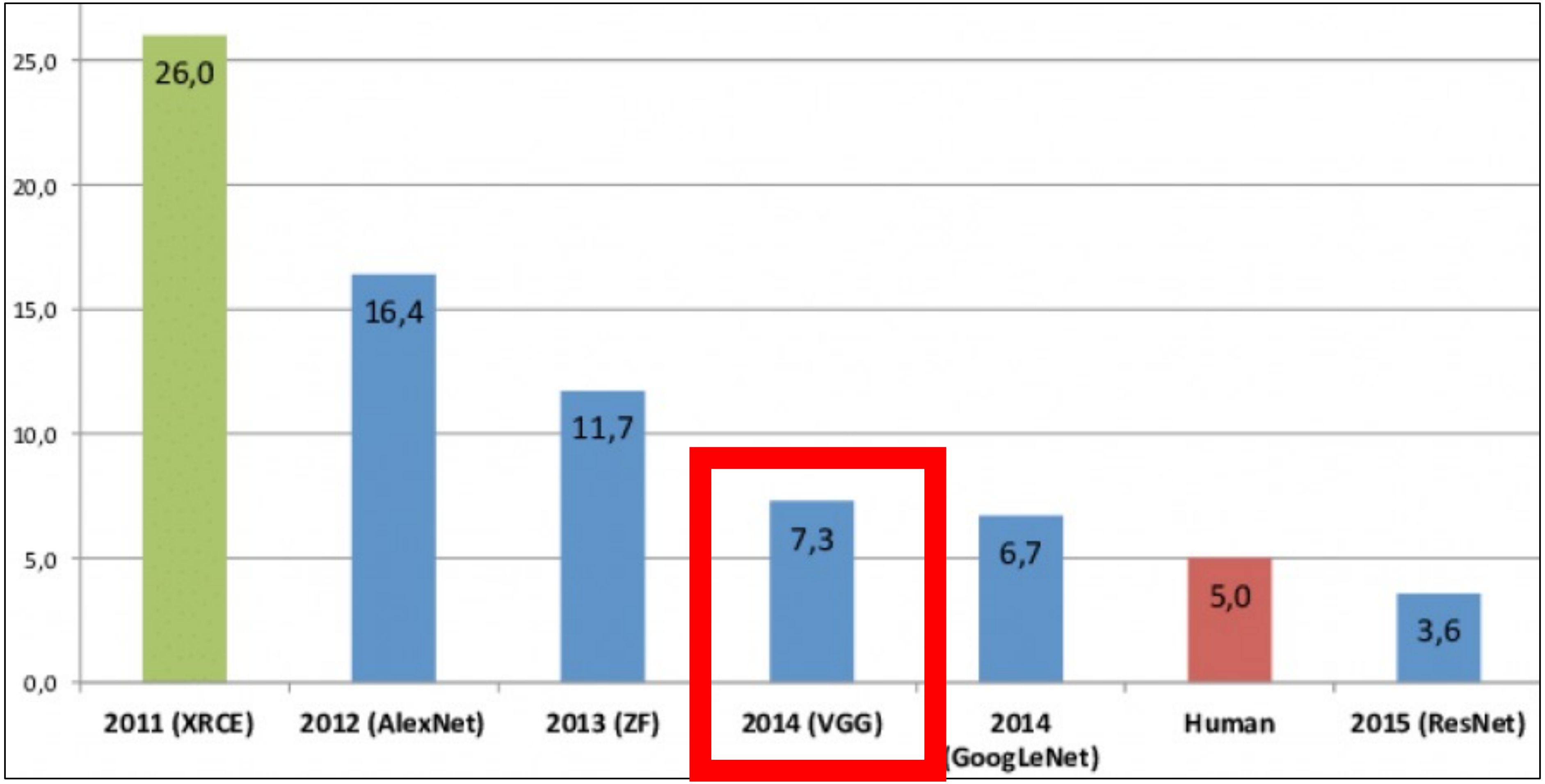


Figure Source: https://www.edge-ai-vision.com/2018/07/deep-learning-in-five-and-a-half-minutes/

Why VGGNet?

VGG stands for the **Visual Geometry Group (VGG)** at University of Oxford where the authors were based ☺

Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." International Conference on Learning Representations (ICLR), 2015.

Key Novelty: **Deeper** Does Better

** Number of layers with learnable model parameters between input and output layer (i.e., excludes pooling layers)*

Figure Source (edited to fix mistakes): https://medium.com/deep-learning-g/cnn-architectures-vggnet-e09d7fe79c45

Key Novelty: **Deeper** Does Better

Figure Source (edited to fix mistakes): https://medium.com/deep-learning-g/cnn-architectures-vggnet-e09d7fe79c45

Key Novelty: **Deeper** Does Better

** Number of layers with learnable model parameters between input and output layer (i.e., exclude pooling layers)*

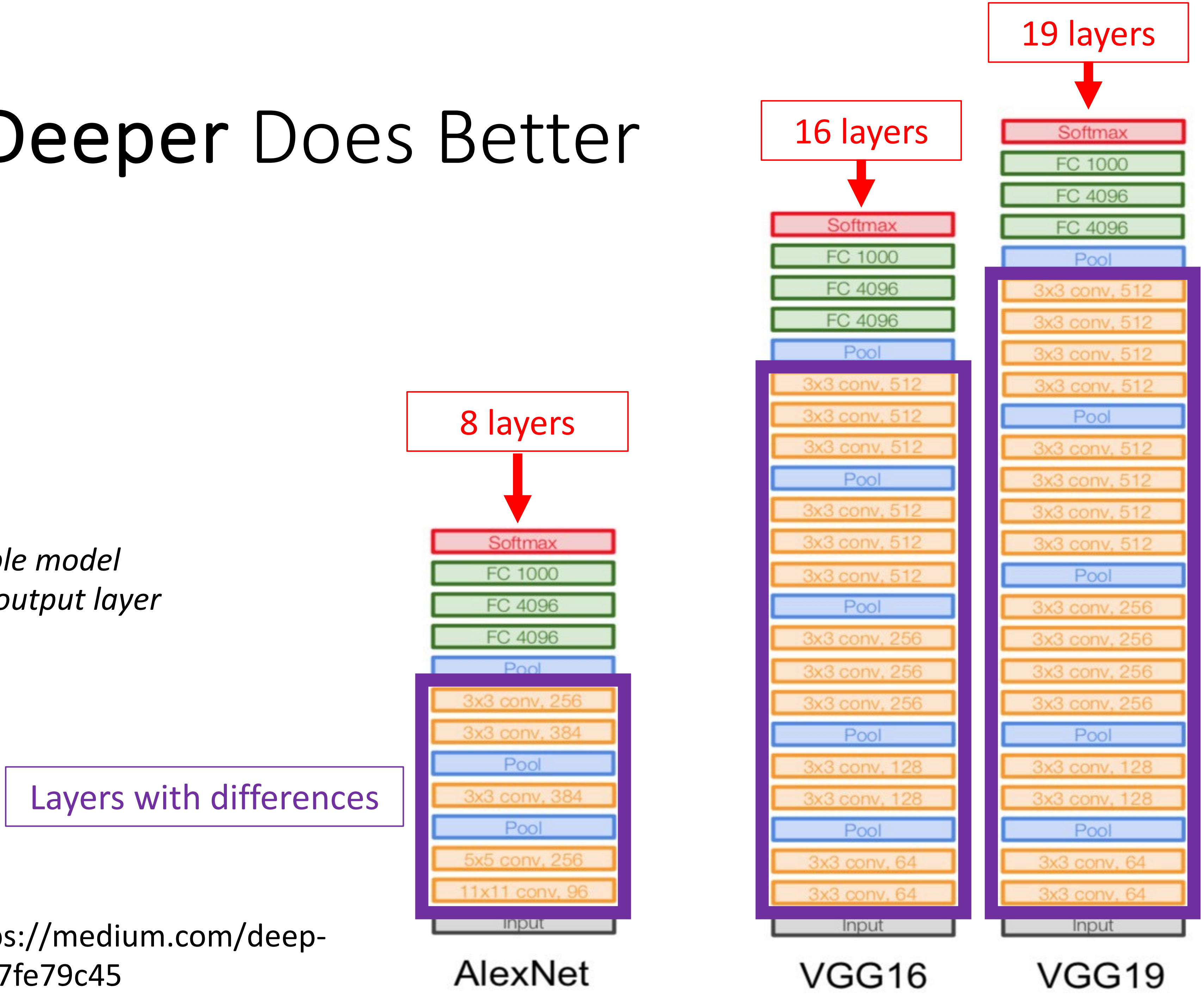


Figure Source (edited to fix mistakes): https://medium.com/deep-learning-g/cnn-architectures-vggnet-e09d7fe79c45

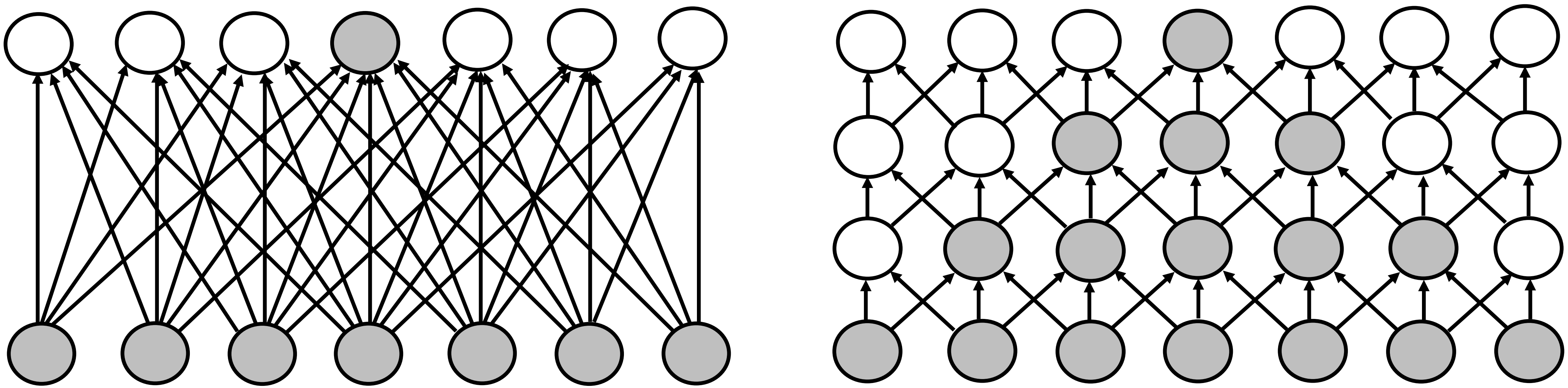
Key Idea: Smaller Convolutional Filters

- Replace larger filter with stack of smaller filters

Figure Source (edited to fix mistakes): https://medium.com/deep-learning-g/cnn-architectures-vggnet-e09d7fe79c45

Key Idea: Smaller Convolutional Filters

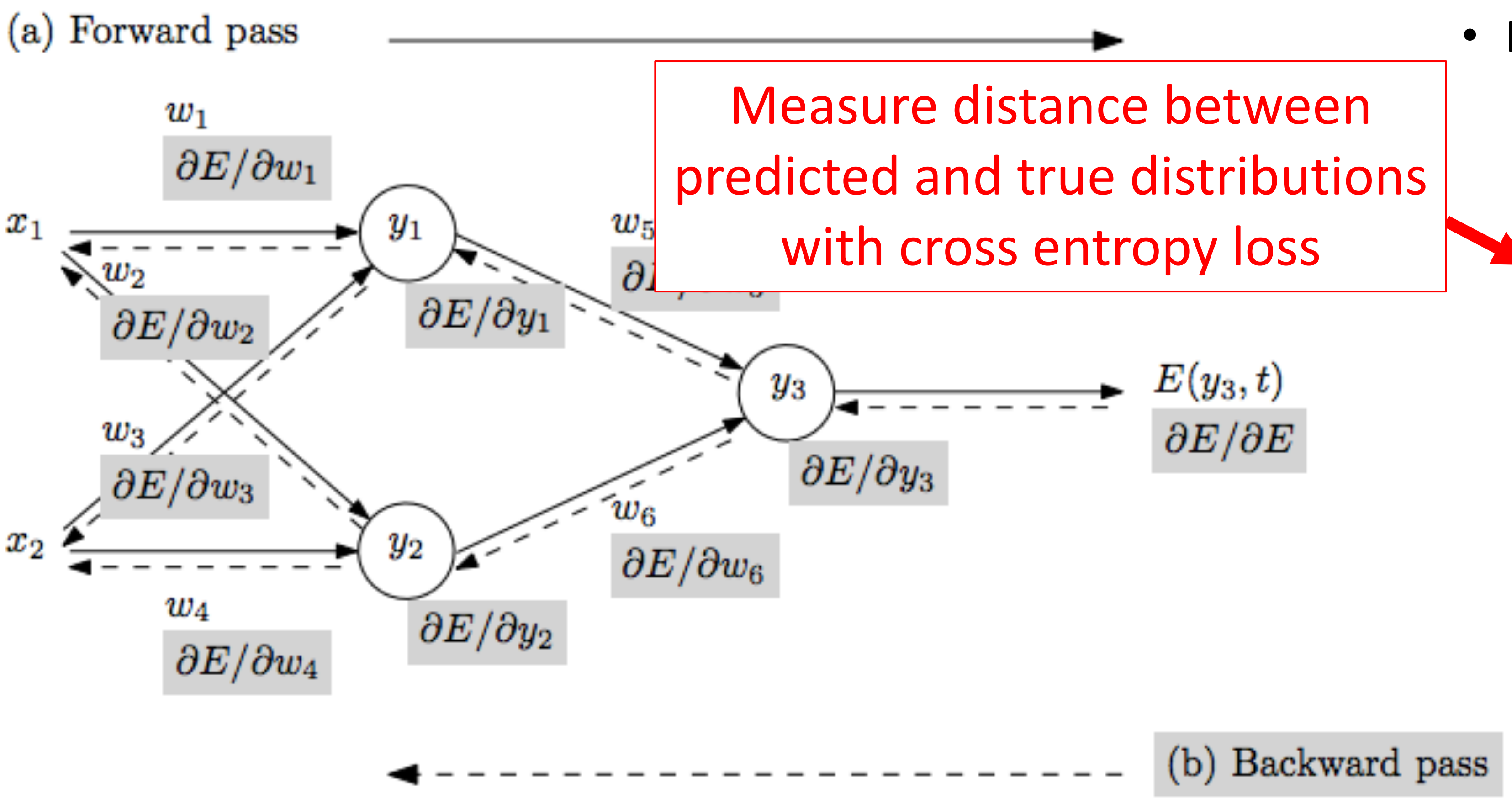
- Replace larger filter with stack of smaller filters; e.g., replace 7x7 with three 3x3s



- Benefits:
  - More discriminative classifier since more non-linear rectifications: 3 vs 1
  - Reduces # of parameters: multiple of 27 (3×3^2) parameters vs 49 (7×7) parameters

https://www.deeplearningbook.org/contents/convnets.html

VGG Training (follows AlexNet): 74 Epochs



Measure distance between predicted and true distributions with cross entropy loss

- Repeat until stopping criterion met:
  1. **Forward pass**: propagate training data through model to make predictions
  2. **Error quantification**: measure dissatisfaction with a model's predictions on training data
  3. **Backward pass**: using predicted output, calculate gradients backward to assign blame to each model parameter
  4. Update each parameter using calculated gradients

https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/

Algorithm Training (follows AlexNet)

- Strategies to mitigate overfitting
    1. Data augmentation
        1. Random patches and their mirror images (2048x more data)
        2. Adjust RGB channels (using PCA to add multiples of principal components)
    2. Dropout (50% of nodes for first two fully connected layers); mimics ensembles by learning to solve same problem with different subnetworks

VGG Limitation: Models Are Large!

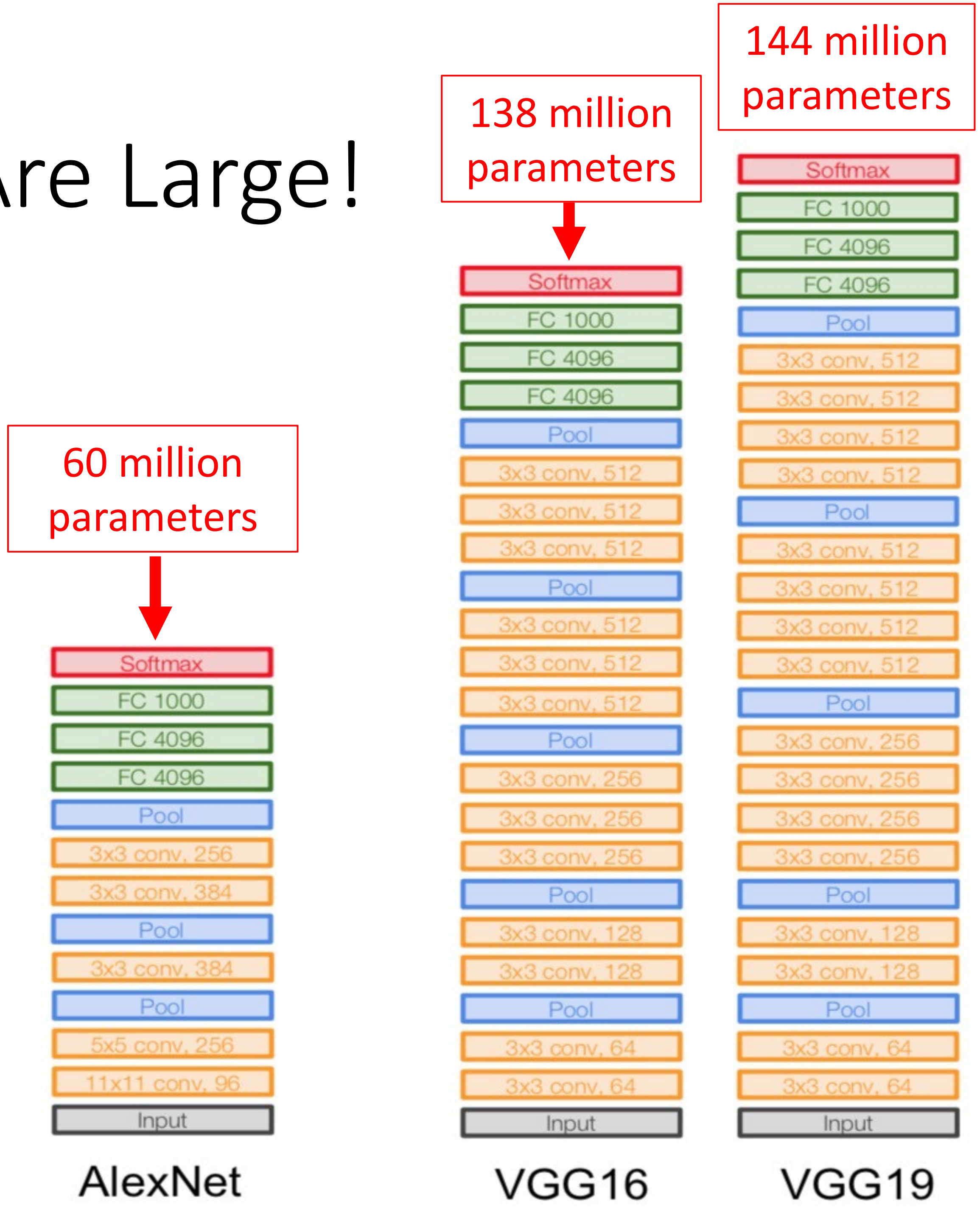
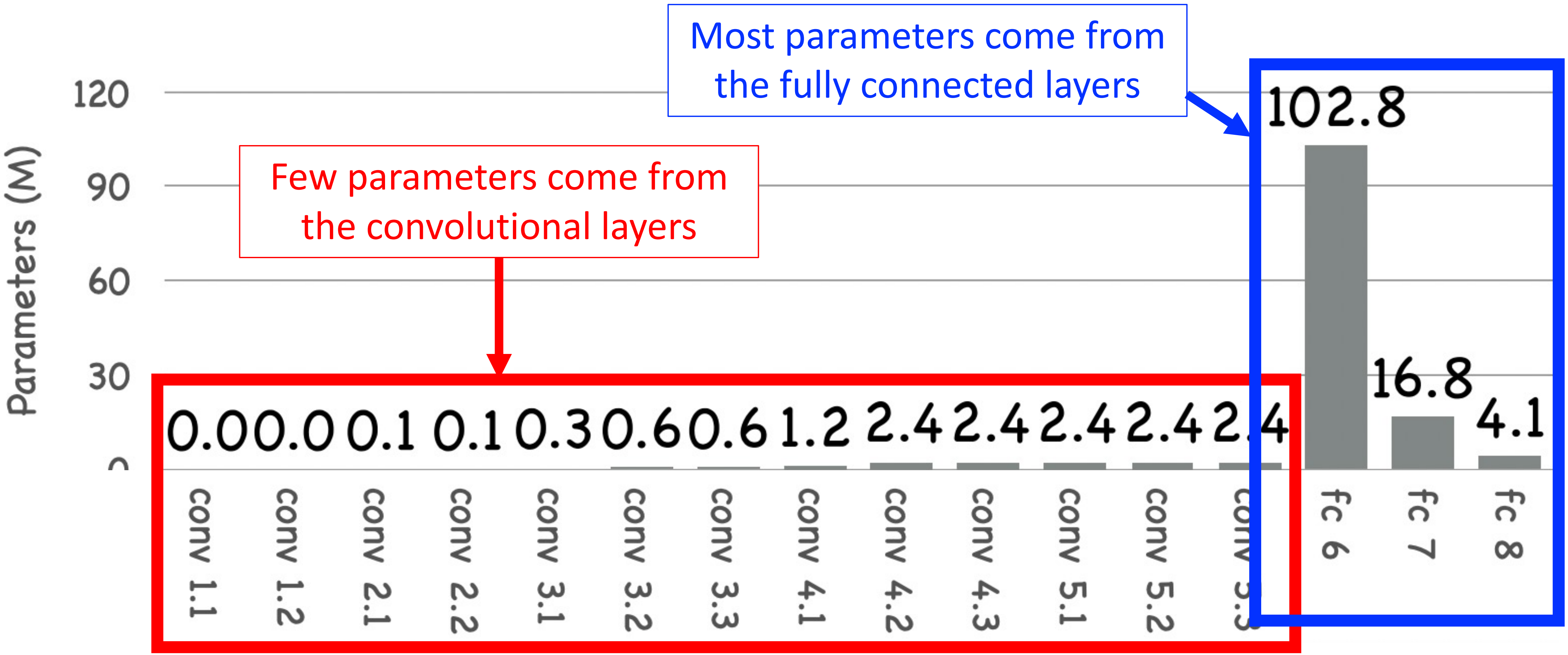


Figure Source (edited to fix mistakes): https://medium.com/deep-learning-g/cnn-architectures-vggnet-e09d7fe79c45

VGG Limitation: Models Are Large (e.g., VGG16)



Source: http://www.philkr.net/cs342/lectures/computer_vision/03.pdf

VGG: Key Tricks for Going Deeper

- 3x3 filters instead of larger filters

- Regularization techniques; e.g.,
    1. Data augmentation
    2. Dropout in fully connected layers

- Trained across multiple GPUs

Object Recognition: Today's Topics

- ImageNet Challenge Top Performers
- Baseline Model: AlexNet
- VGG
- ResNet
- Discussion

Secret Sauce for State-of-Art: Deeper CNNs

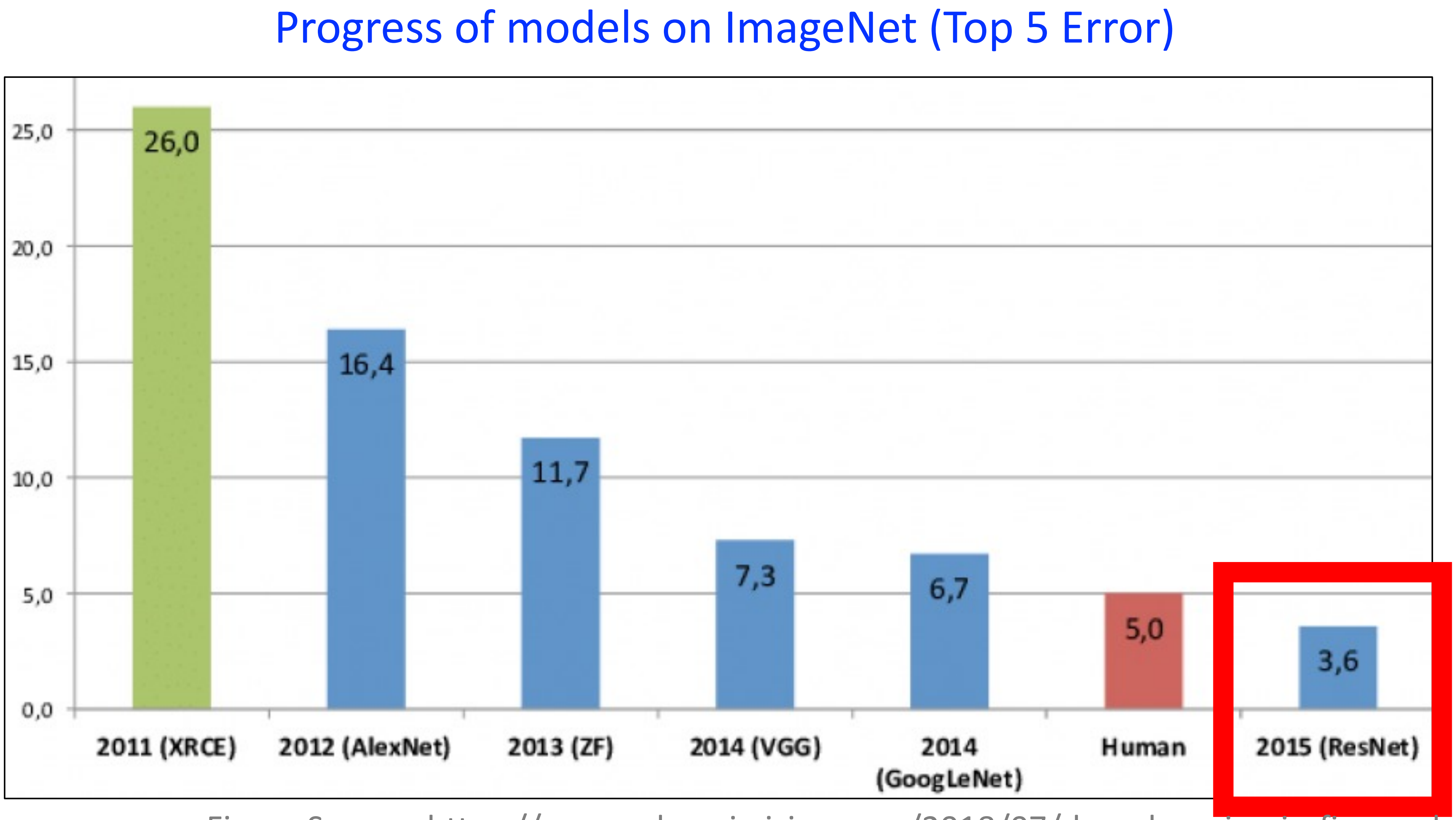


Figure Source: https://www.edge-ai-vision.com/2018/07/deep-learning-in-five-and-a-half-minutes/

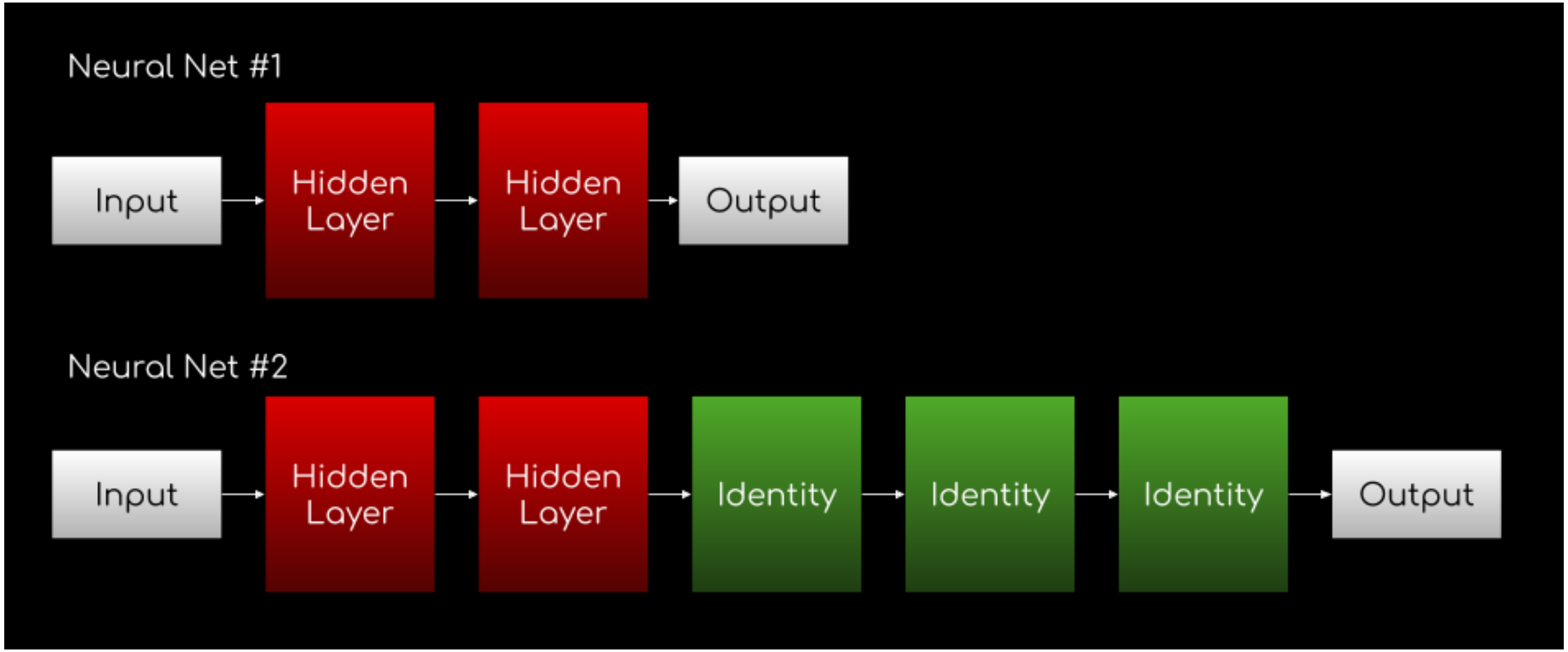
Why ResNet?

"Res" stands for residuals, which is the key novel idea in the proposed algorithm.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." CVPR, 2016.

Motivating Observation

Idea: a deeper network should perform as good if not better than shallower networks since they can learn the shallower function by simply learning “identity” functions for later layers
Observation: adding more layers leads to WORSE results!
Is the problem overfitting?



Source: https://medium.com/@realmichaelye/intuition-for-resnet-deep-residual-learning-for-image-recognition-39d24d173e78

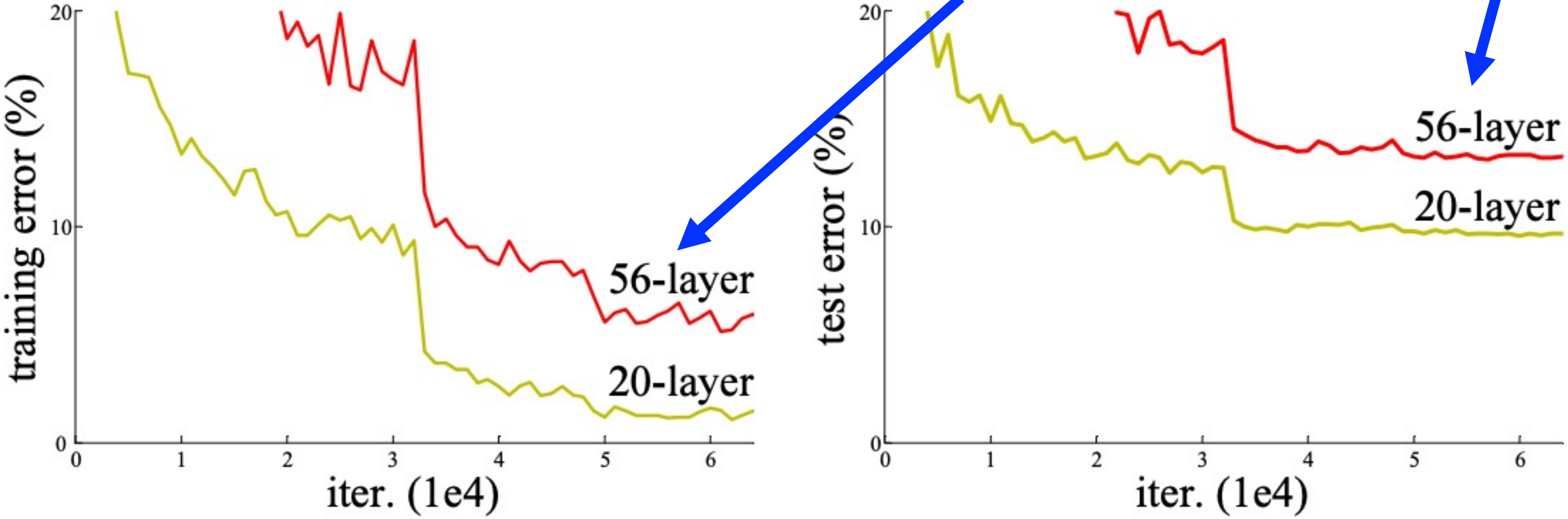
Motivating Observation

Idea: a deeper network should perform as good if not better than shallower networks since they can learn the shallower function by simply learning “identity” functions for later layers

Observation: adding more layers leads to WORSE results!

Is the problem overfitting? NO

Training data error (and test error) is greater with more layers



Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition.” CVPR, 2016.

Motivating Observation

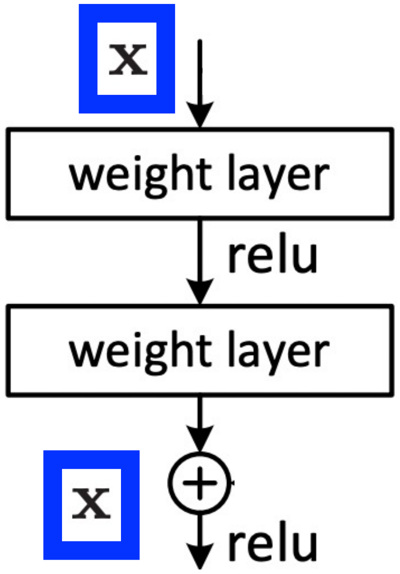
Idea: a deeper network should perform as good if not better than shallower networks since they can learn the shallower function by simply learning “identity” functions for later layers
Observation: adding more layers leads to WORSE results!
Is the problem overfitting? NO

Problem: It is difficult for the algorithm to learn layers of identity mappings

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. ”Deep Residual Learning for Image Recognition." CVPR, 2016.

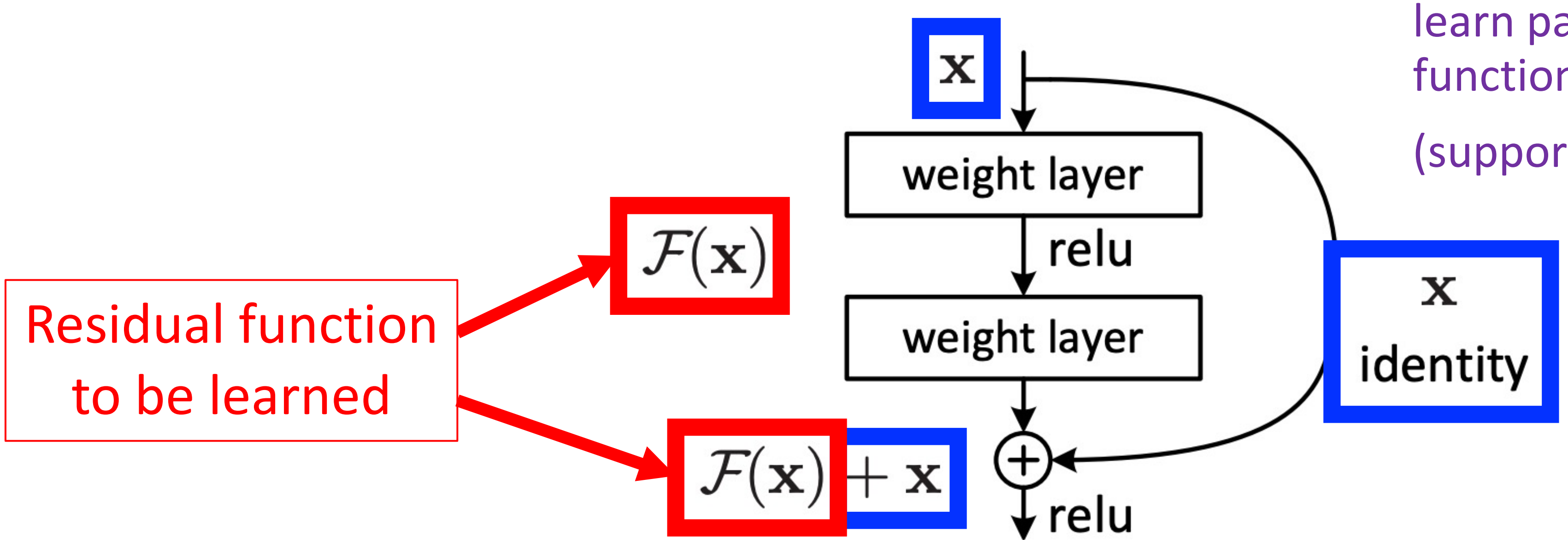
Problem: Difficult to Perform Identity Mapping

e.g.,



Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." CVPR, 2016.

Key Idea: Skip Connections that Perform Identity Mapping



Assumption: Easier for a network to learn parameters close to 0 than a function that performs identity mapping

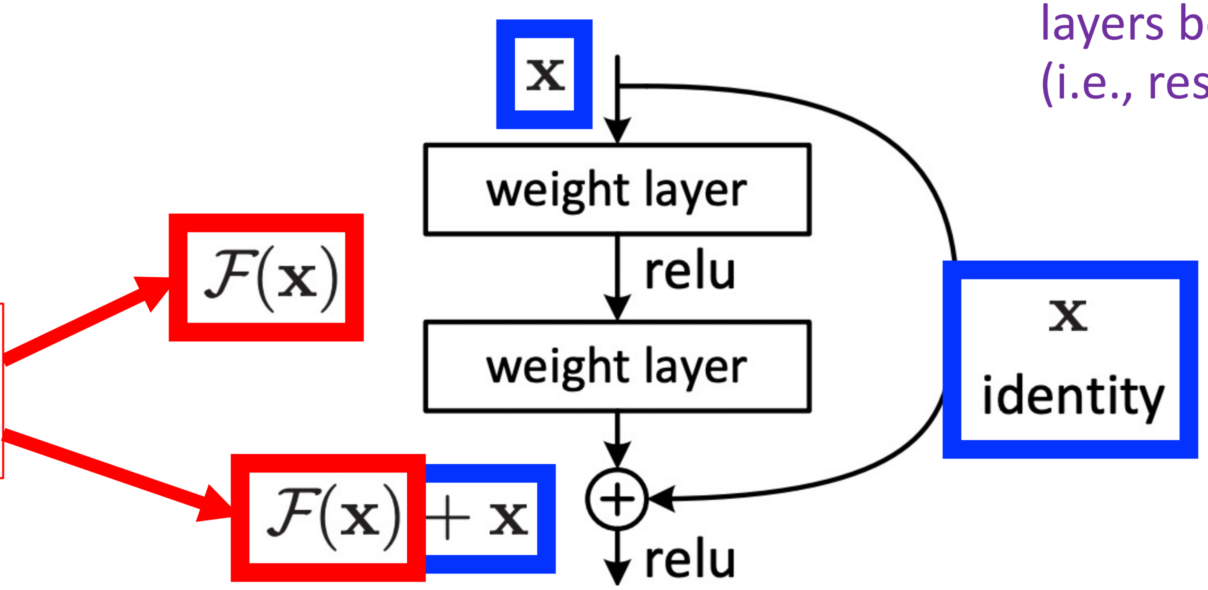
(supported by experiments in the paper)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." CVPR, 2016.

Key Idea: Skip Connections that Perform Identity Mapping

Paper explores when number of layers between skip connections (i.e., residual block) is 2 or 3

Residual function to be learned



Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." CVPR, 2016.

Key Idea: Skip Connections that Perform Identity Mapping

How many model parameters need to be learned when adding a skip connection?

- 0

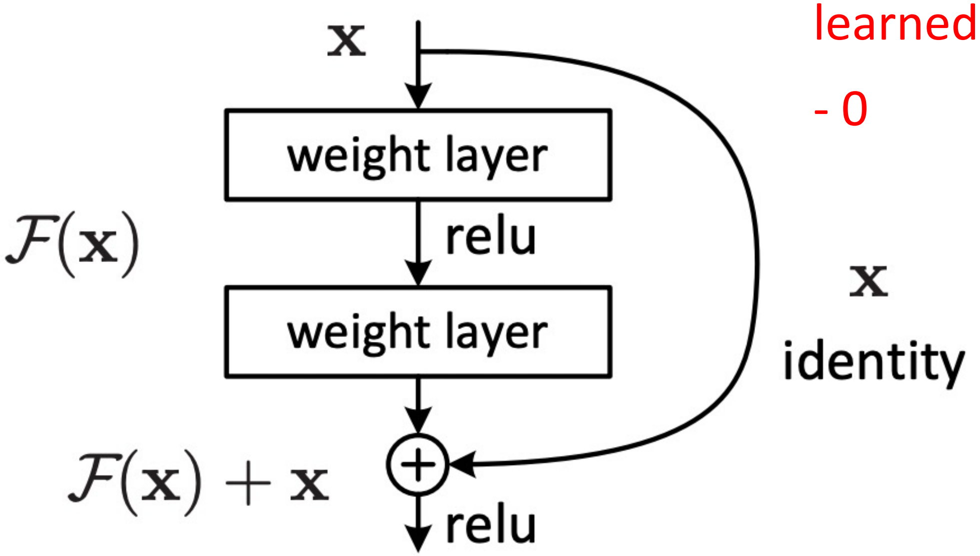
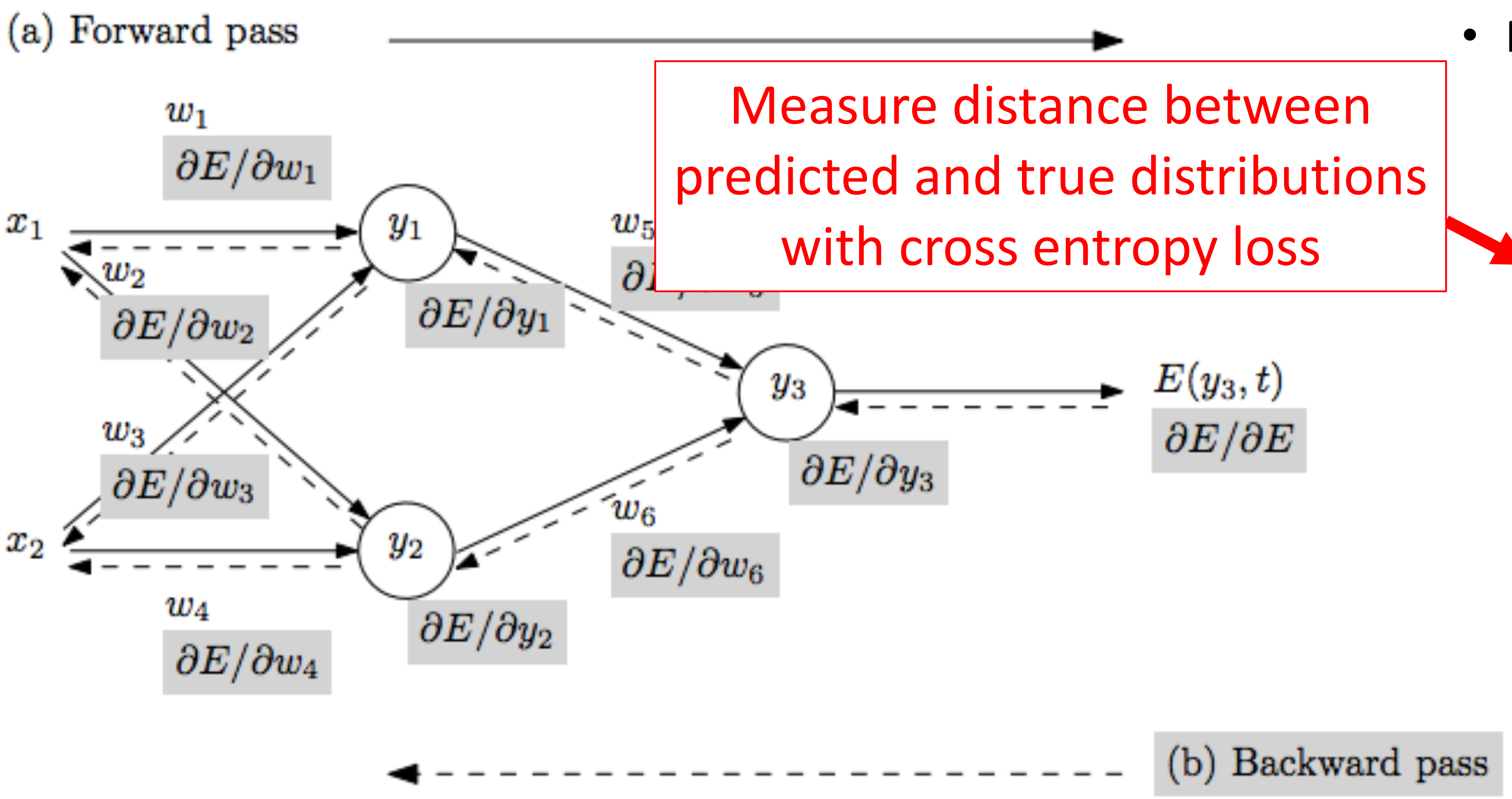


Figure 2. Residual learning: a building block.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." CVPR, 2016.

ResNet Training (follows AlexNet)



Measure distance between predicted and true distributions with cross entropy loss

- Repeat until stopping criterion met:
  1. **Forward pass**: propagate training data through model to make predictions
  2. **Error quantification**: measure dissatisfaction with a model's predictions on training data
  3. **Backward pass**: using predicted output, calculate gradients backward to assign blame to each model parameter
  4. Update each parameter using calculated gradients

https://www.jefkine.com/general/2016/09/05/backpropagation-in-convolutional-neural-networks/

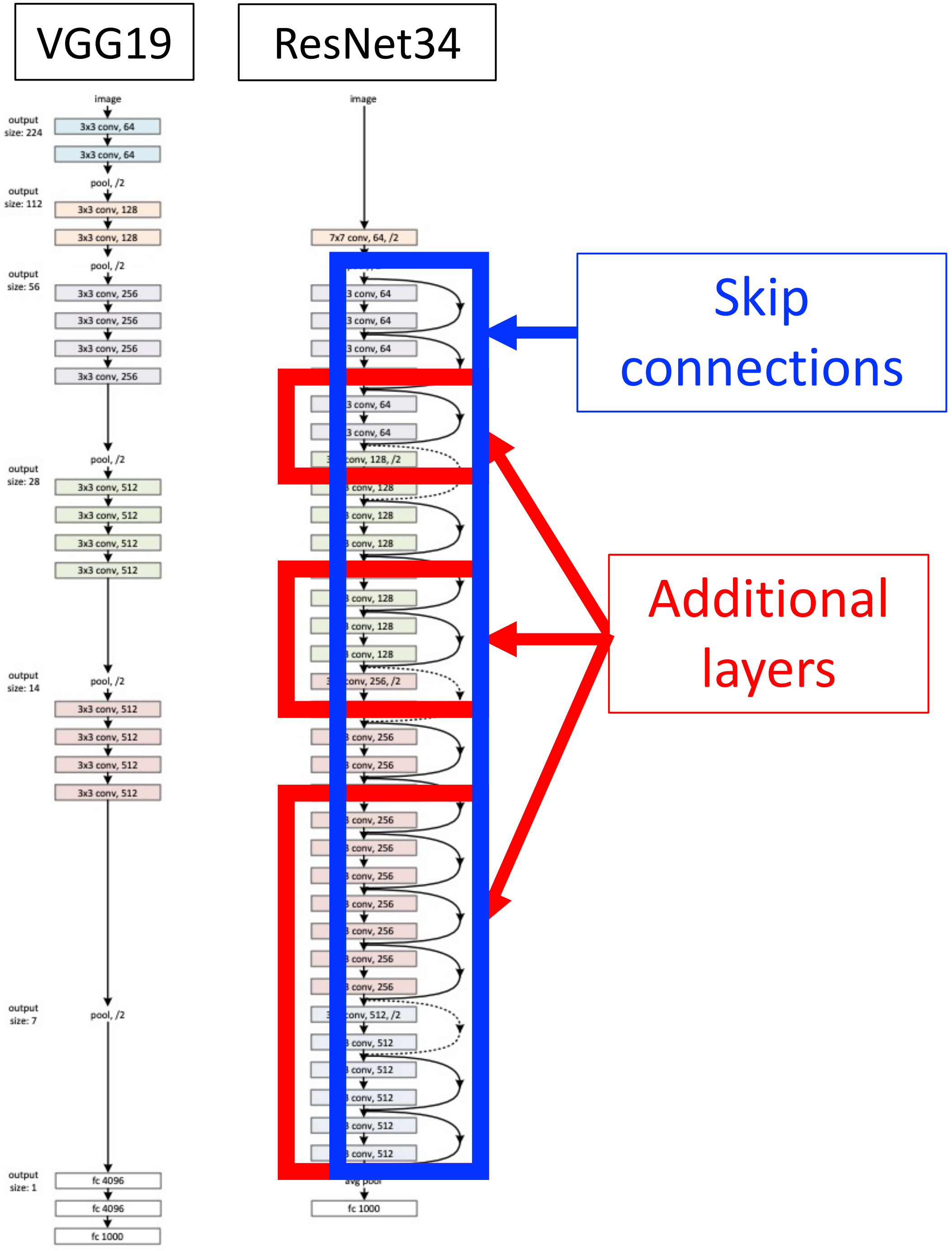
ResNet Training (follows AlexNet)

- Strategy to mitigate overfitting
    1. Data augmentation
        1. Random patches and their mirror images (2048x more data)
        2. Adjust RGB channels (using PCA to add multiples of principal components)

ResNet Implementations

Deep residual learning framework using skip connections enabled successfully learning deeper models than prior work

(18, 34, 50, 101, & 152 layers!)



Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. ”Deep Residual Learning for Image Recognition." CVPR, 2016.

Experimental Results on Validation Set

model	top-1 err.	top-5 err.
VGG-16 [40]	28.07	9.33
GoogLeNet [43]	-	9.15
PReLU-net [12]	24.27	7.38
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Performance improves with more layers

ResNet models outperform prior state-of-art models!

ResNet: Key Tricks for Going Deeper

- Skip connections with residual learning

Object Recognition: Today’s Topics

- ImageNet Challenge Top Performers
- Baseline Model: AlexNet
- VGG
- ResNet
- Discussion

State-of-Art Design Models Go “Deeper”

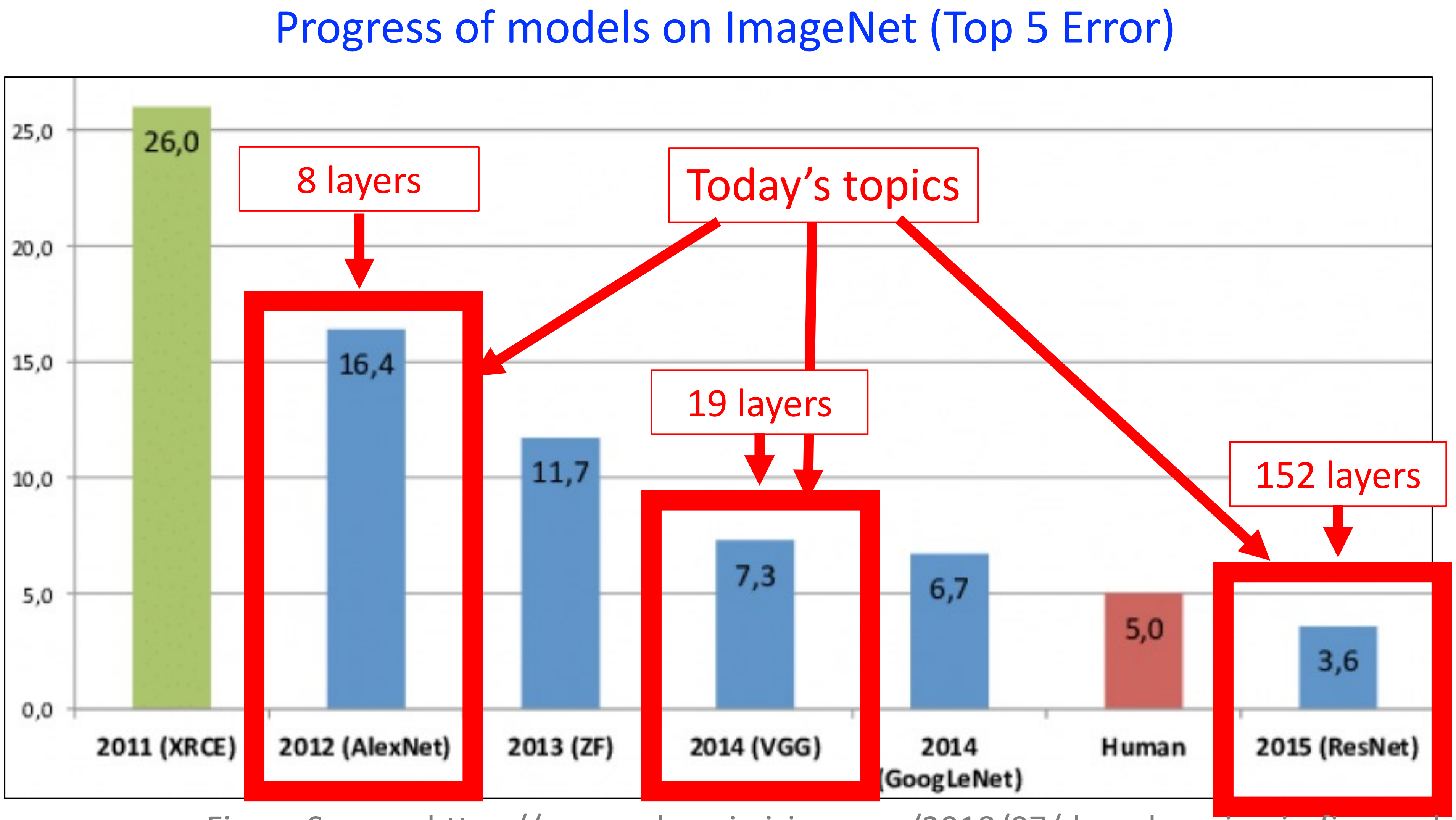


Figure Source: https://www.edge-ai-vision.com/2018/07/deep-learning-in-five-and-a-half-minutes/

State-of-Art Model Exceeds Human Performance!

Progress of models on ImageNet (Top 5 Error)

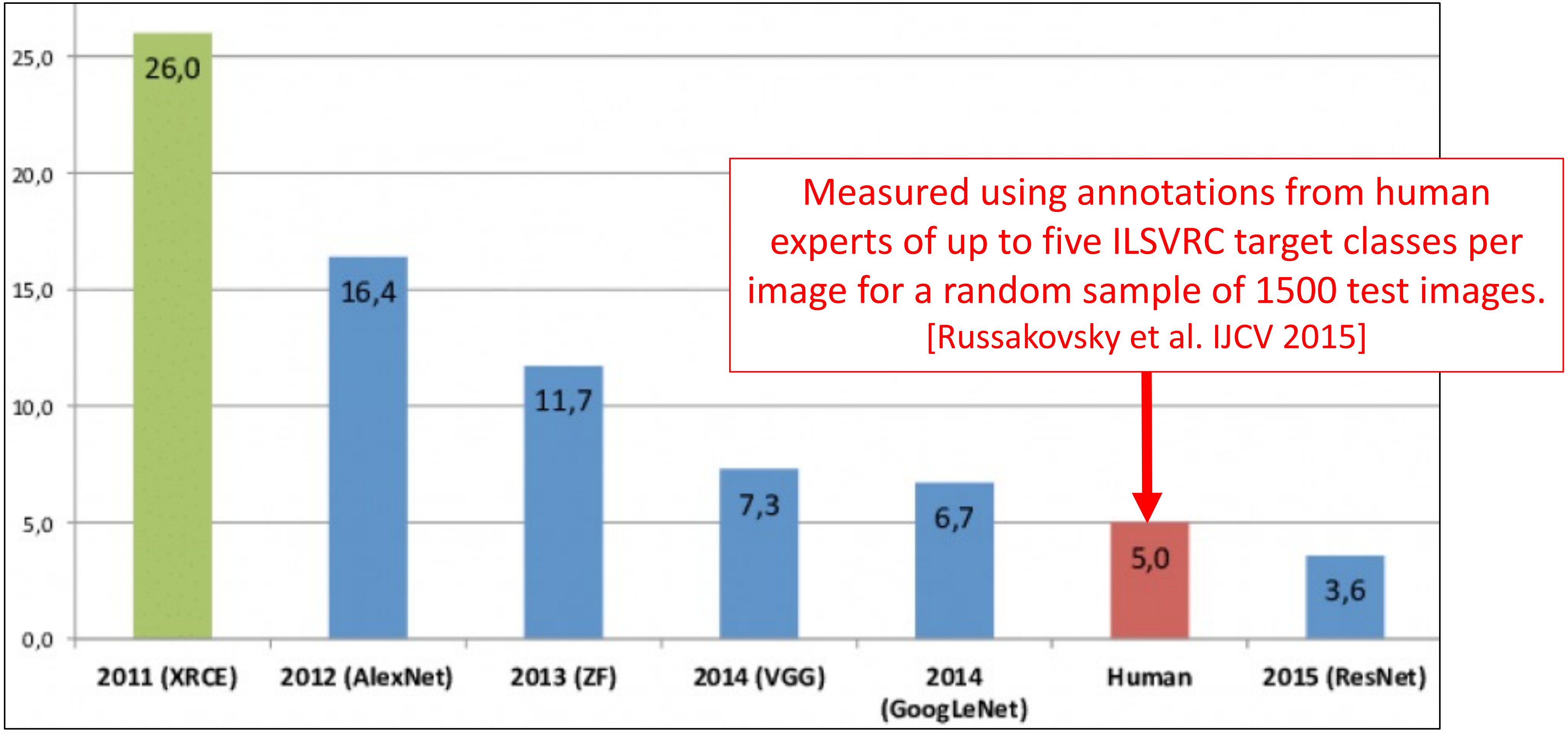
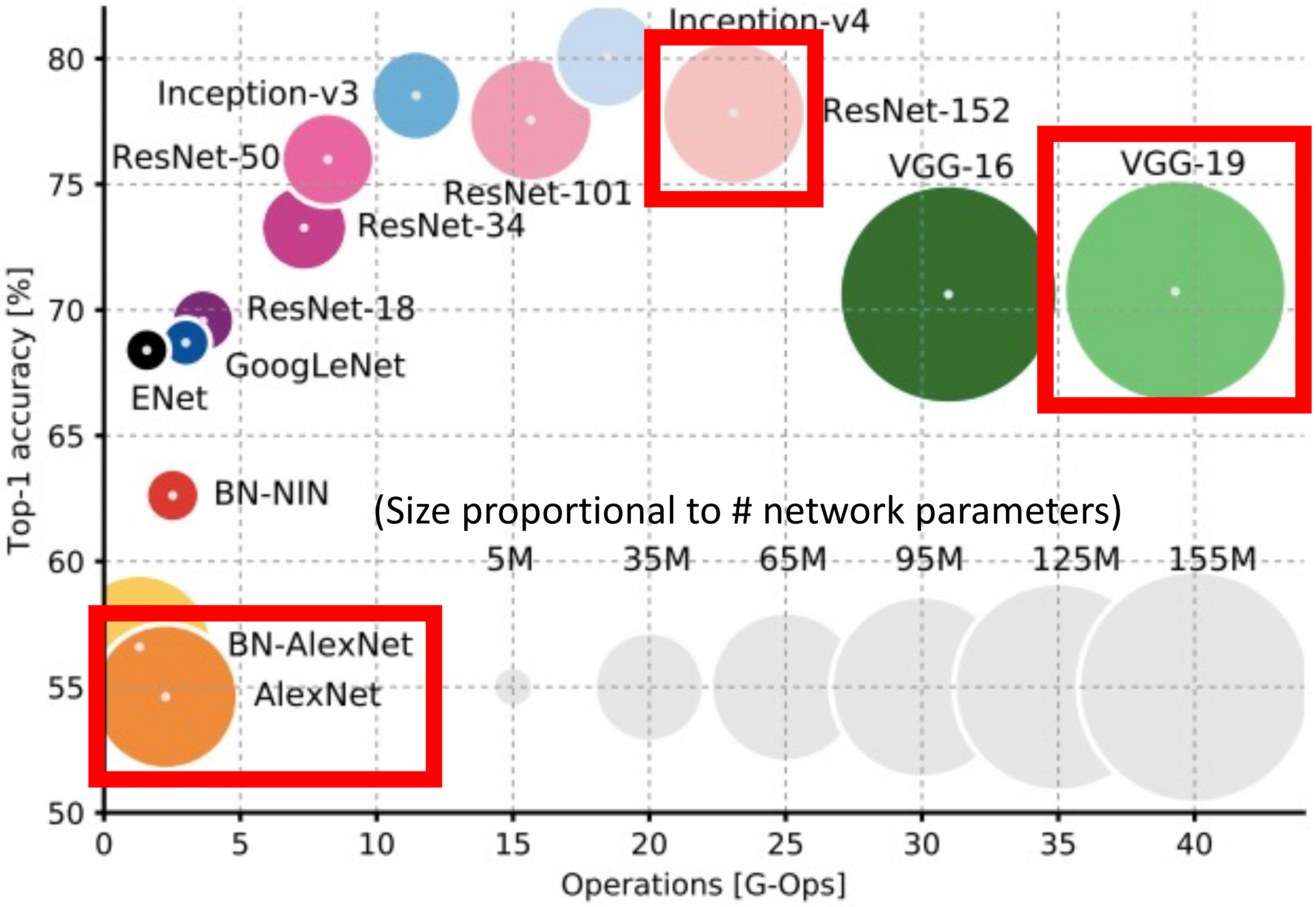


Figure Source: https://www.edge-ai-vision.com/2018/07/deep-learning-in-five-and-a-half-minutes/

CNN Architectures Are a Great Start… Transformers to Follow



(required for a single forward pass)

https://arxiv.org/pdf/1605.07678.pdf

Group Discussion

- Vote for today’s topics in the Google form:

Object Recognition: Today's Topics

- ImageNet Challenge Top Performers
- Baseline Model: AlexNet
- VGG
- ResNet
- Discussion

Object Recognition: Today’s Topics

- ImageNet Challenge Top Performers
- Baseline Model: AlexNet
- VGG
- ResNet
- Discussion


The End