- Welcome to the Deep Learning Workshop by HKSAIR!

- **Session 1: Diving into CNN**
  - Review: Linear Model and non-linear extension
  - The evolution of different CNN network architectures

- **Coffee break**
  - Please remember to register the lab platform

- **Session 2: Hands-on Lab Tutorial**
  - TensorFlow 2.0
  - Recognize the hand-written characters

Diving into CNN

Wang Weiyan HKUST
wwangbc@cse.ust.hk

Acknowledge Alex Lo & Bingxiao Shi
for collecting materials and drawing figures

Outline

Hong Kong Society of Artificial Intelligence and Robotics 香港人工智能與機器人學會

- Review: Linear Model
- Non-linear Classification:
  - Width: Kernel Trick and SVM
  - Depth: Deep Neural Network
- CNN milestone Works:
  - Lenet
  - Alexnet
  - VGG
  - …
  - SENet
- Common Practice: Finetuning(Transfer Learning)

Review: Logistic Regression

- **Linear Regression**: the start point of all stories

$$y = f(x) = \sum_{i=0}^{D} w_i x_i = w^T x$$

$$\min_{w} L(w) = \frac{1}{N} \sum_{i=0}^{N} (\mathrm{y} - w^T x)^2$$

- **Classification**: logistic Regression replace y with $\ln(\frac{y}{1-y})$

$$\ln\left(\frac{y}{1-y}\right) = f(x) = w^T x \Rightarrow y = \frac{1}{1+e^{-w^T x}}$$

Kernel: Infinite Width Extension

Hong Kong Society of Artificial Intelligence and Robotics 香港人工智能與機器人學會

- **Polynomial Regression**: replace x with non-linear function $\phi(x)$

$$y = f(x) = \sum_{i=0}^{D} w_i \phi(x_i) = w^T \phi(x)$$

where $\phi(x) = [1, x_1, x_2, x_1^2, x_1 x_2, x_2^2, \dots, x_2^d]$

- **Kernel trick**: get infinite width of $\phi(x)$ for SVM

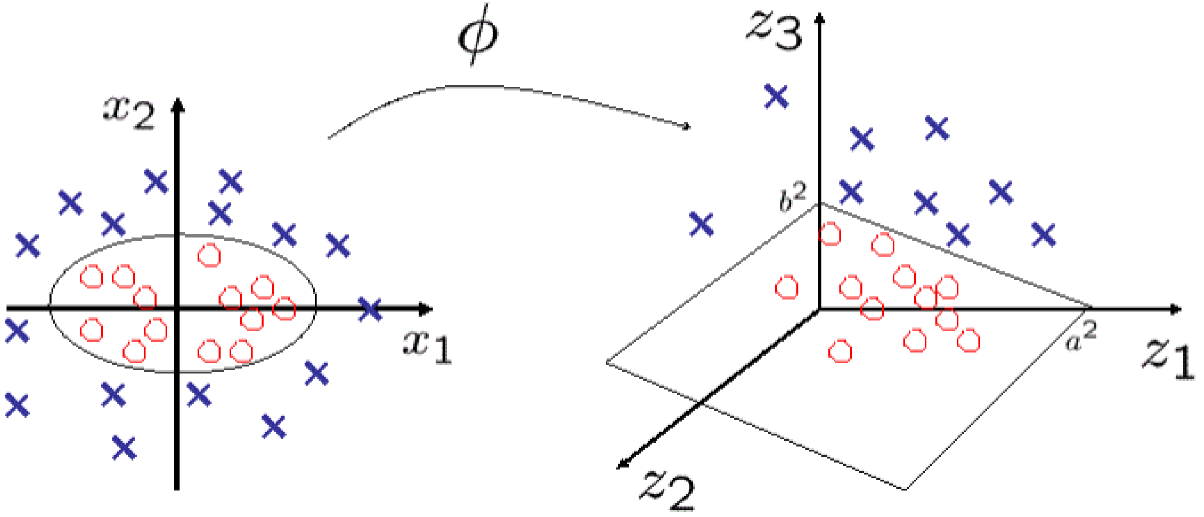
$$k(x, x_i) = \phi(x)\phi(x_i) = \exp(-\frac{|x - x_i|^2}{2\sigma^2})$$

Similarity!

$$y = w^T k(x,\cdot) = \sum_{i=0}^{N} \alpha_i y_i \phi^T(x_j)\phi(x) = \sum_{i=0}^{N} \alpha_i y_i \exp(-\frac{|x - x_i|^2}{2\sigma^2})$$

Kernel: Infinite Width Extension

Nonlinear model



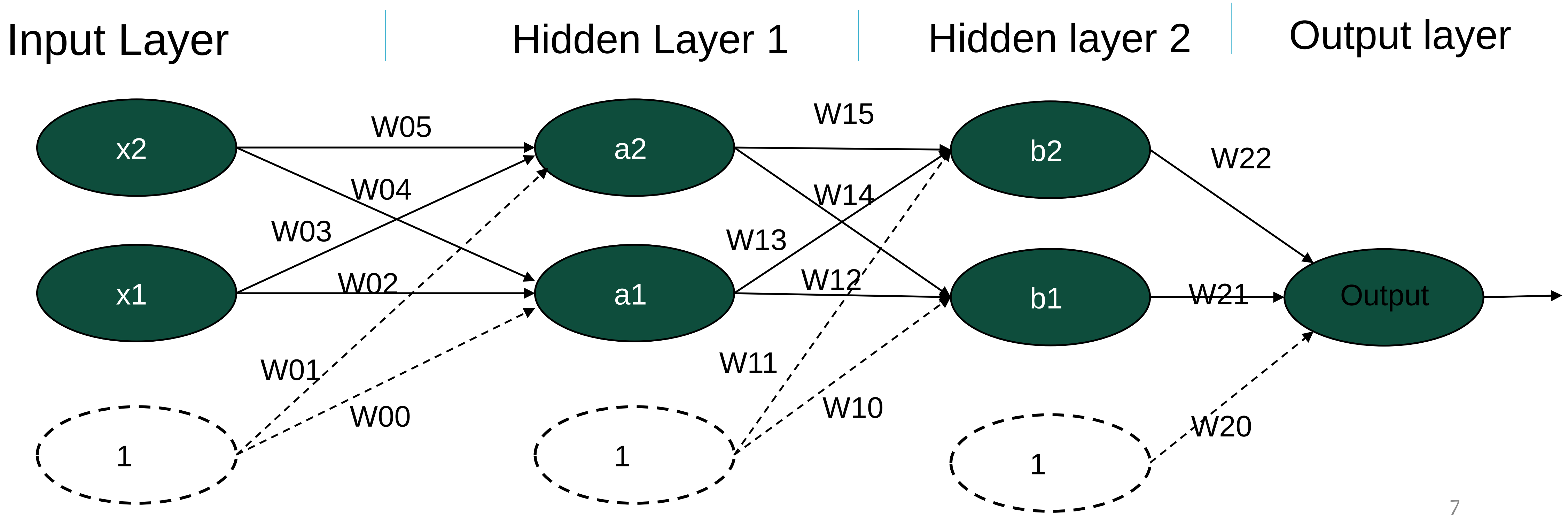
build hypothesis $\longrightarrow$ $y = w^T\phi(x)$

Linear model

Vladimir N. Vapnik

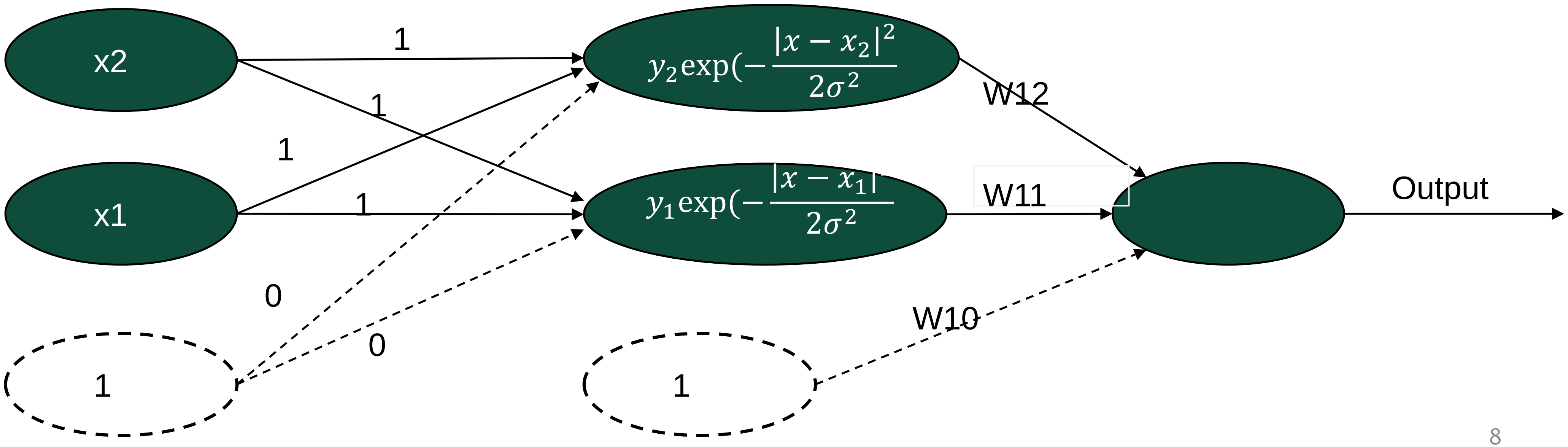
Deep NN: Depth Extension

- Multilayer Perceptron (MLP)
  - Stacking logistic regression
  - The hidden layer output is the result of one linear splitting
  - Activation function: logistic function (tanh, Relu …)



Kernel Trick VS. DNN

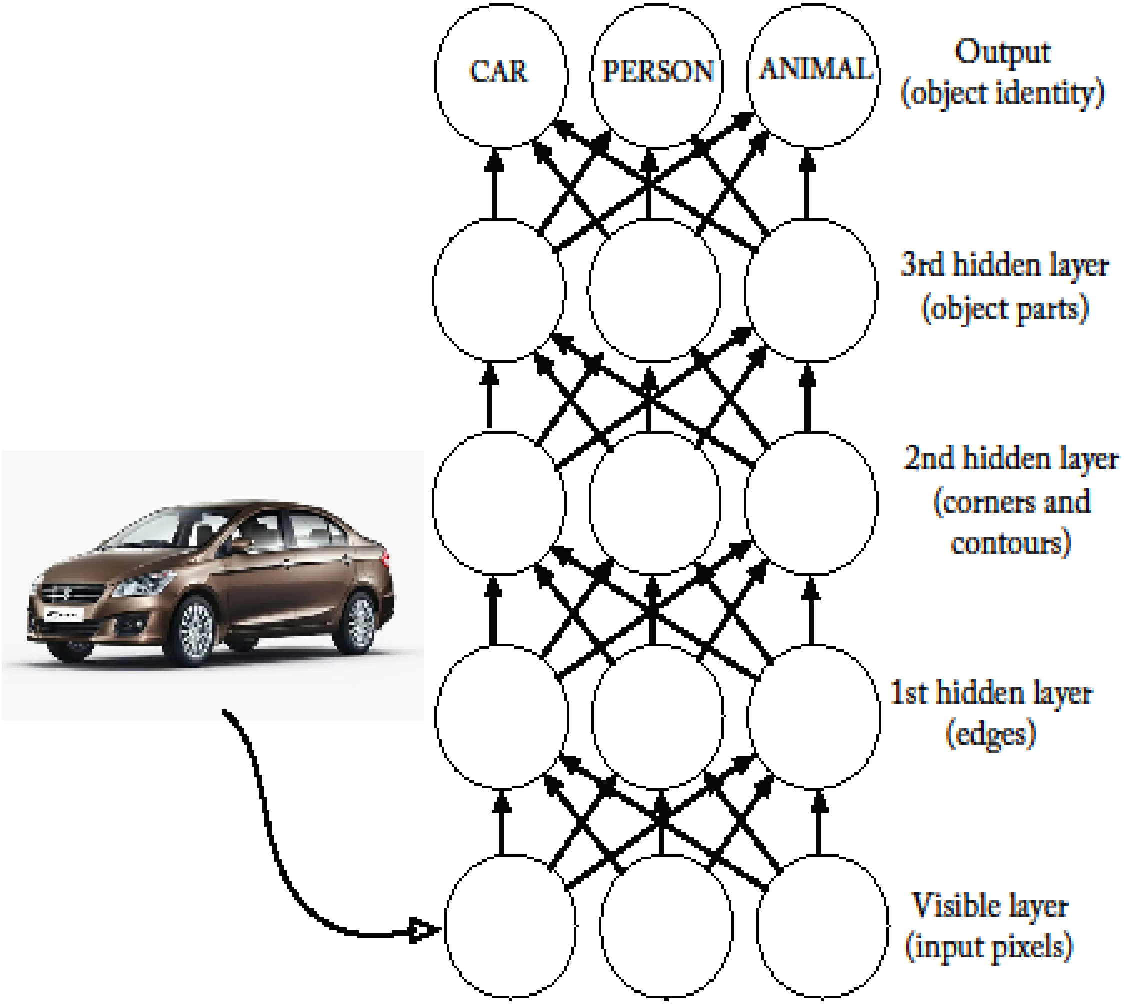
- Kernel Trick is a special case of DNN
  - One hidden layer with fixed weights
  - Hidden layer: Nodes# = Data Sample#
  - Activation: $\exp(-\frac{|x-x_i|^2}{2\sigma^2})$ instead of logistic, tanh or relu



Width VS. Depth

- Deep Learning:
  - Hierarchical Representation

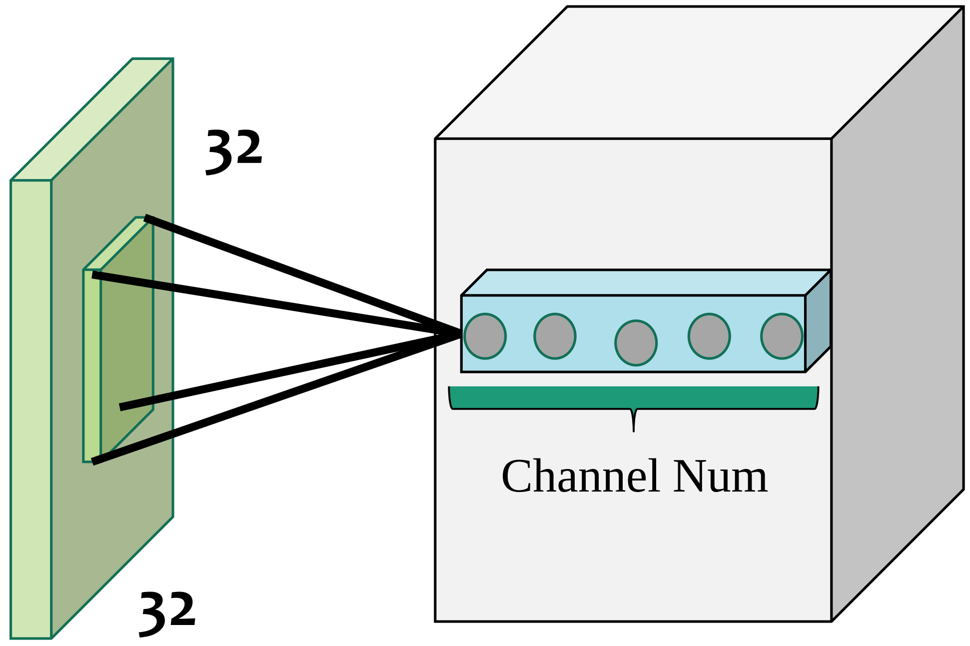
- Intuition: Nodes#
  - Exponential growth for expression ability in depth
  - Linear growth in width
  - Hard to choose a kernel

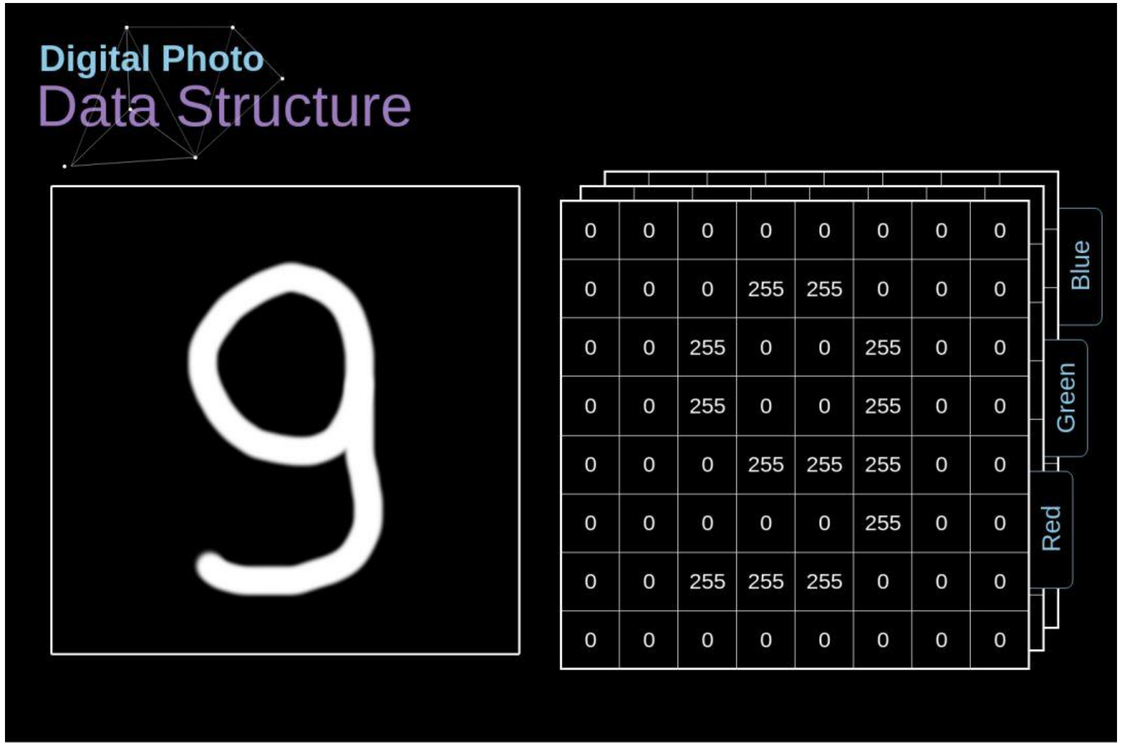


Lenet: Form MLP to CNN

Hong Kong Society of Artificial Intelligence and Robotics
香港人工智能與機器人學會

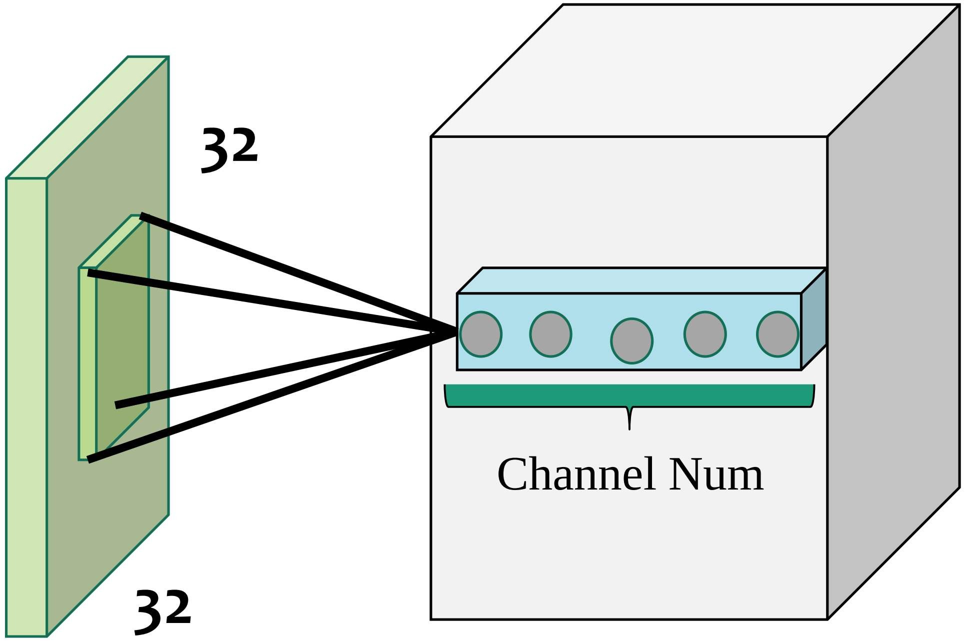
- Convolution Operation: Spatial Localization
  - Kernel Size: the window size the conv cares about
  - Padding: pad the image at the edge to maintain the size
  - Stride Size: the step that the window slides

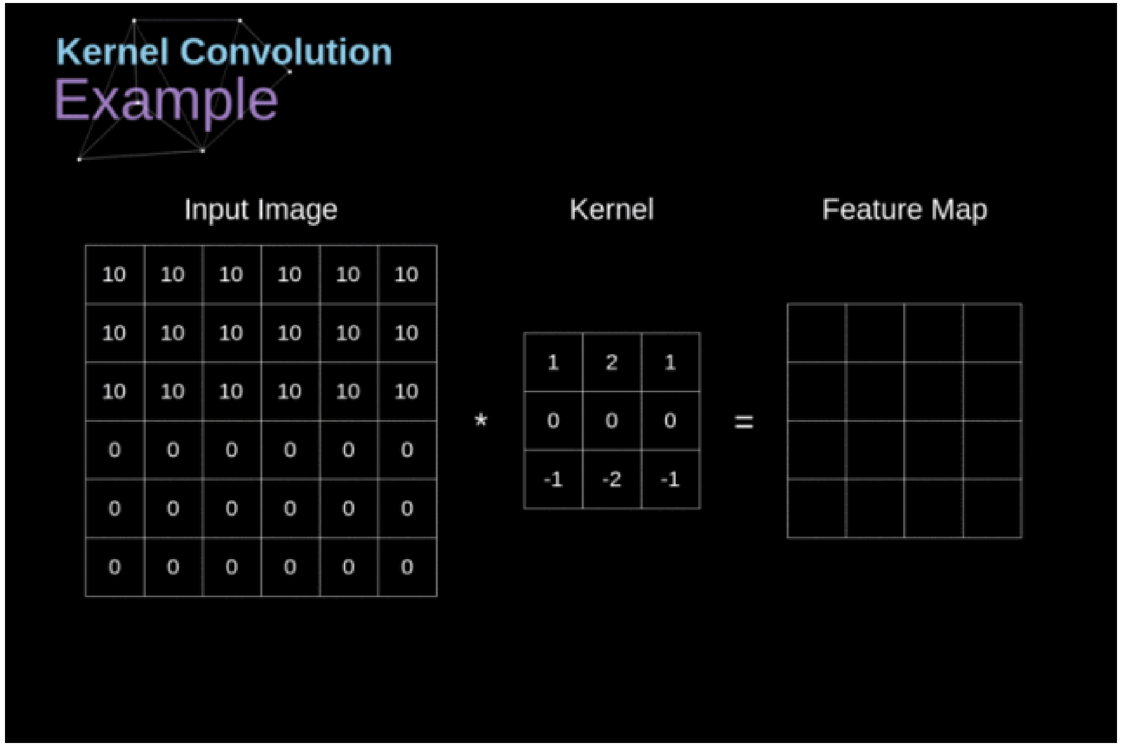




Lenet: Form MLP to CNN

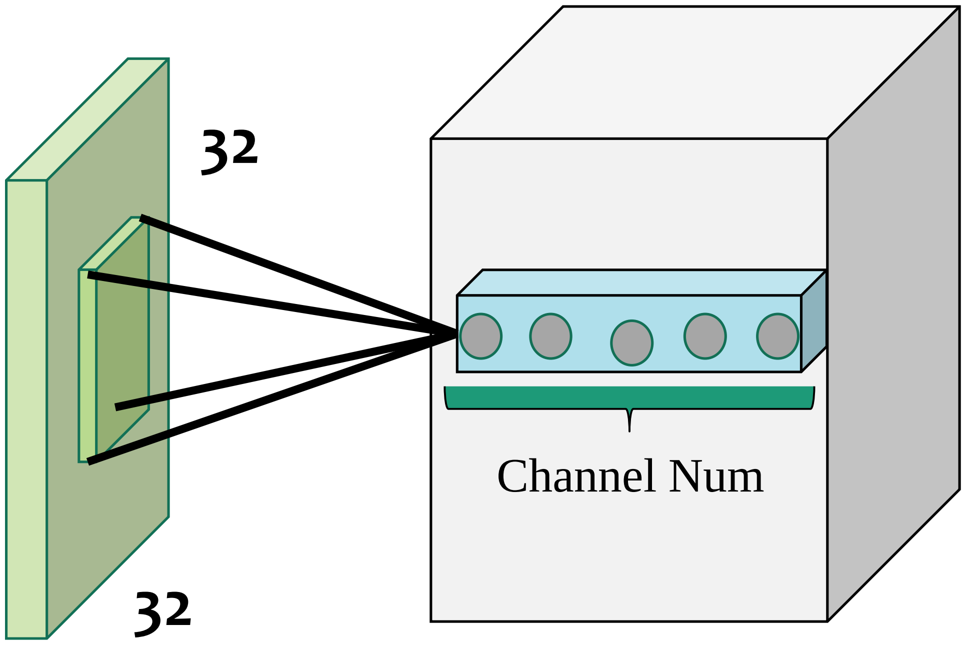
- Convolution Operation: Spatial Localization
  - Kernel Size: the window size the conv cares about
  - Padding: pad the image at the edge to maintain the size
  - Stride Size: the step that the window slides





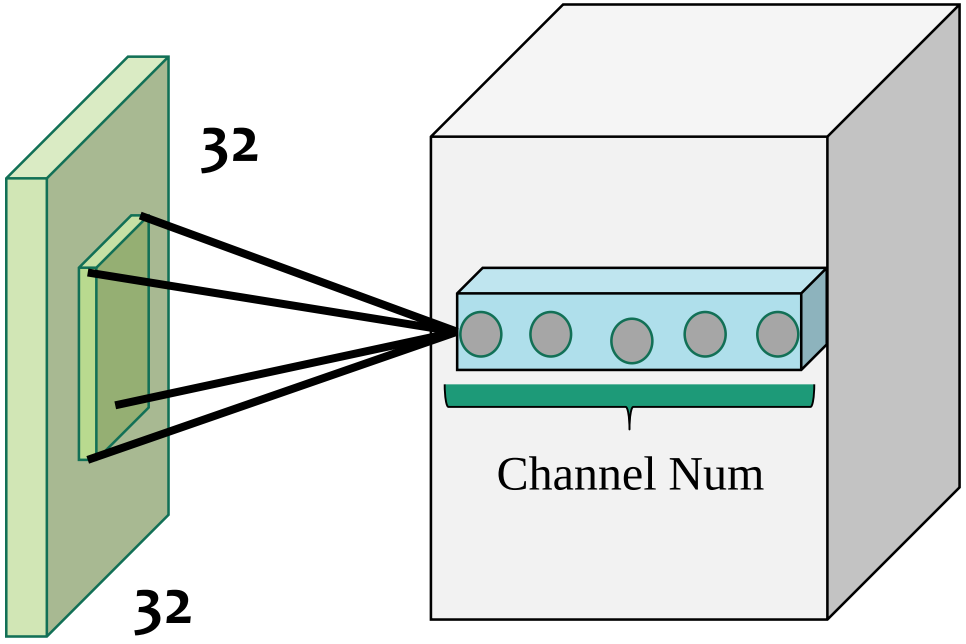
Lenet: Form MLP to CNN

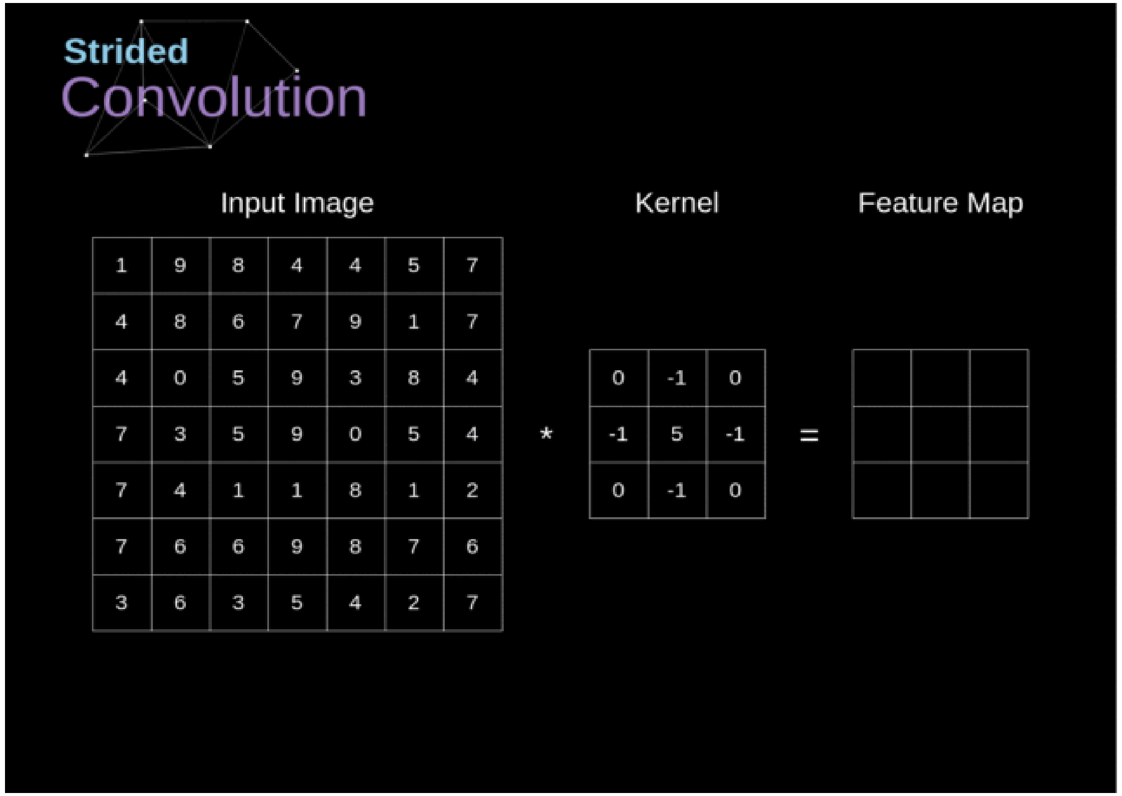
- Convolution Operation: Spatial Localization
  - Kernel Size: the window size the conv cares about
  - Padding: pad the image at the edge to maintain the size
  - Stride Size: the step that the window slides



Lenet: Form MLP to CNN

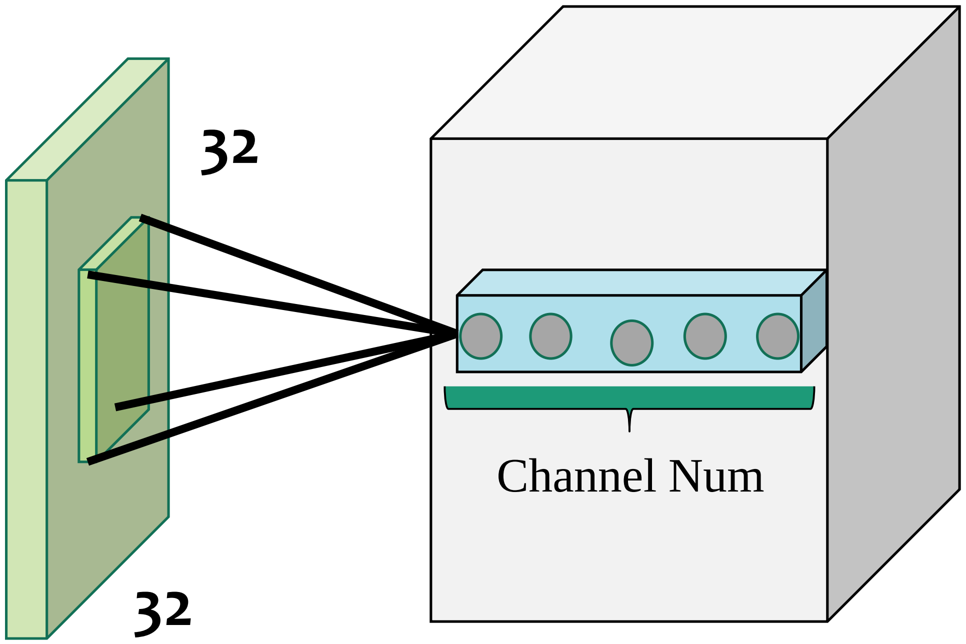
- Convolution Operation: Spatial Localization
  - Kernel Size: the window size the conv cares about
  - Padding: pad the image at the edge to maintain the size
  - Stride Size: the step that the window slides

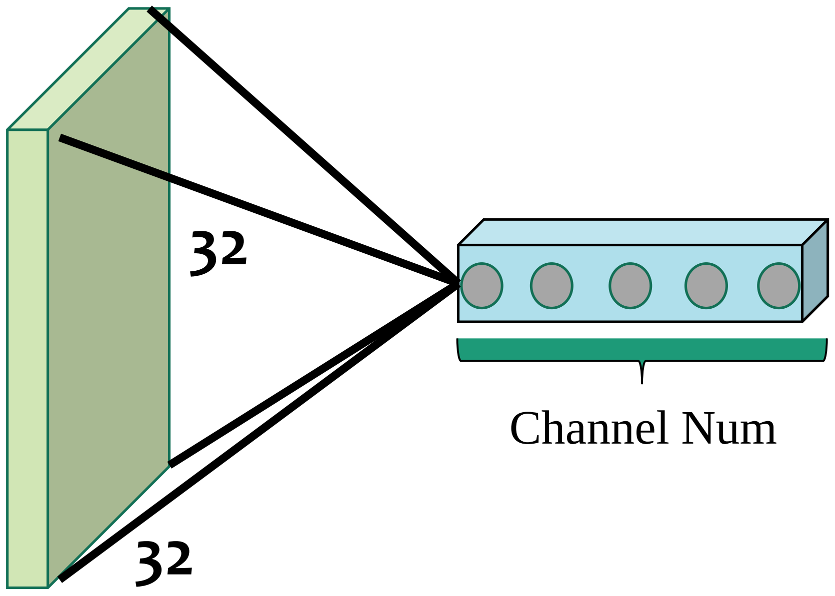




Lenet: Form MLP to CNN

- Dense Layer (Fully Connected Layer)
  - A special case of Convolution
  - Kernel Size = Image Size
  - All linear operations are convolution



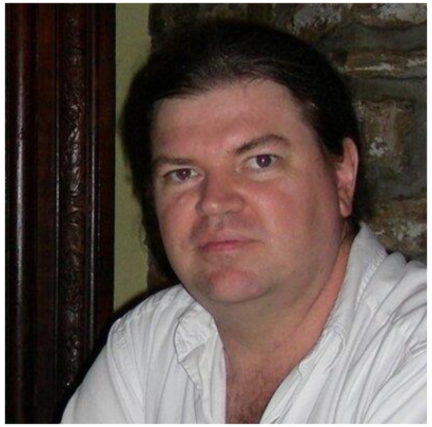


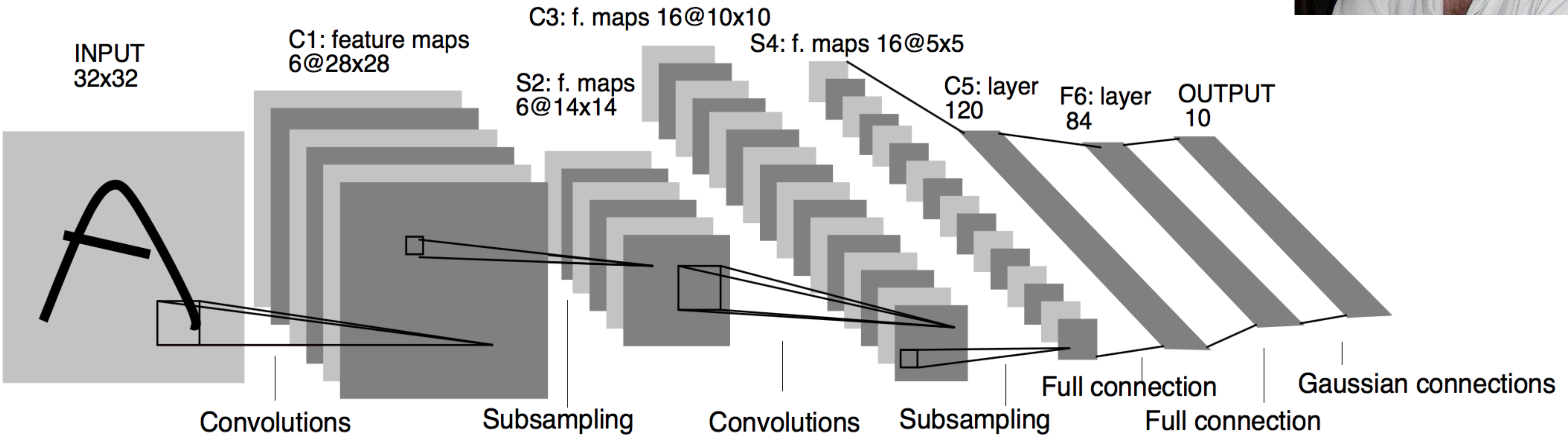
Lenet: Form MLP to CNN

Hong Kong Society of Artificial Intelligence and Robotics 香港人工智能與機器人學會

Convolution Layer/Block:

1. Convolution: local linear operation
2. Activation: Relu, tanh, sigmoid etc…
3. Subsampling(Pooling): Maxpooling, Mean-pooling …

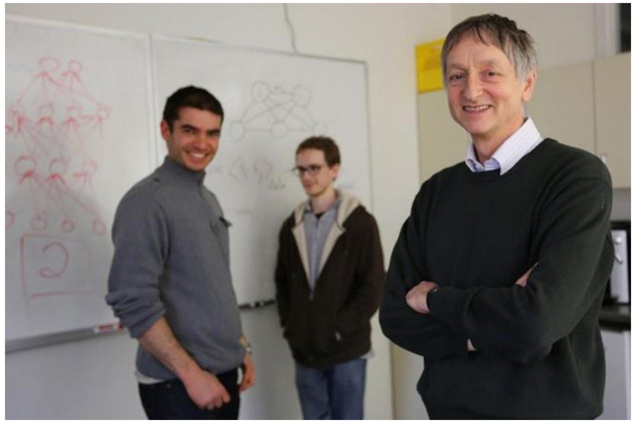


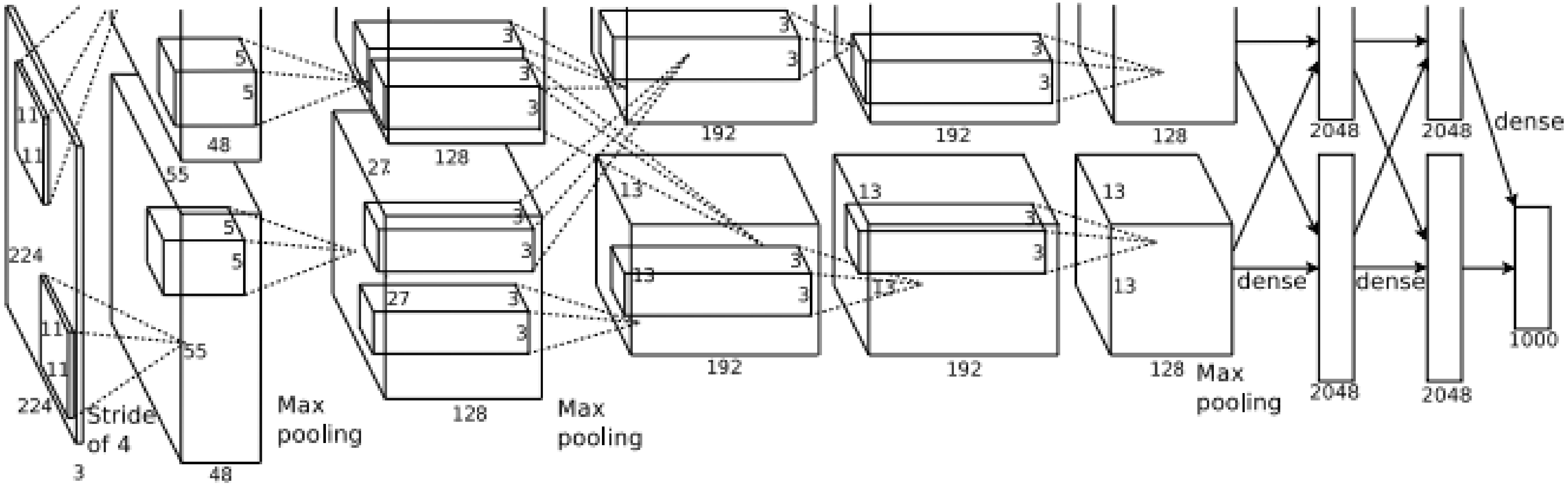


9*9 9*9

Alexnet: dropout

- Alexnet: 7 layers Deep CNN
  - Winner of ImageNet 2012
  - top-5 error: 15.3% (2^{nd} place: ~25%)
  - Start the DL revolution in CV





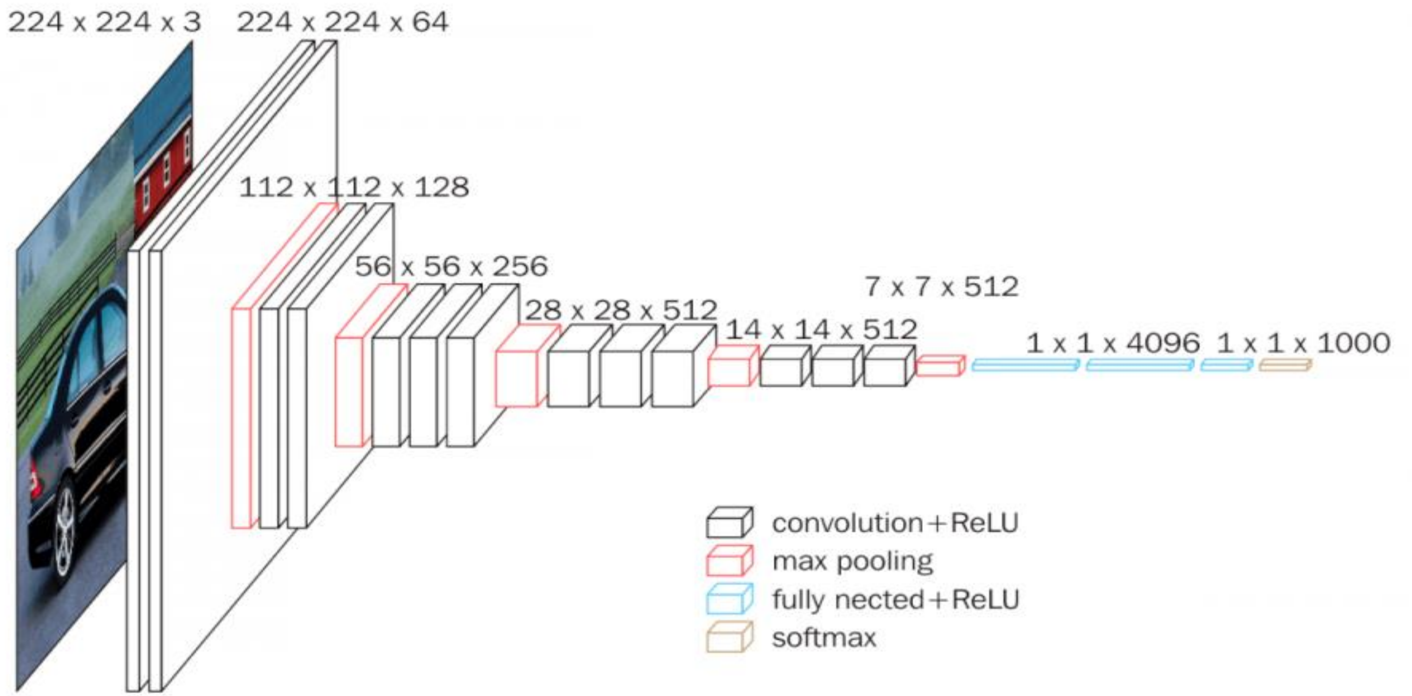
11*11 5*5 3*3 3*3 3*3

Alexnet: Dropout

- Why Alexnet work and what's difference:
  - Large data: ImageNet
  - GPU: accelerate the training
  - Data Augmentation: flipping, rotation, resize, cropping …
  - **Dropout: a pixel of the feature map to be zero with 50% probability**
- **Regularization Interpretation:**
  - Randomly augmented the feature map with occlusion and prevent overfitting
- **Ensemble Interpretation:**
  - Training sub networks and test with all the sub networks

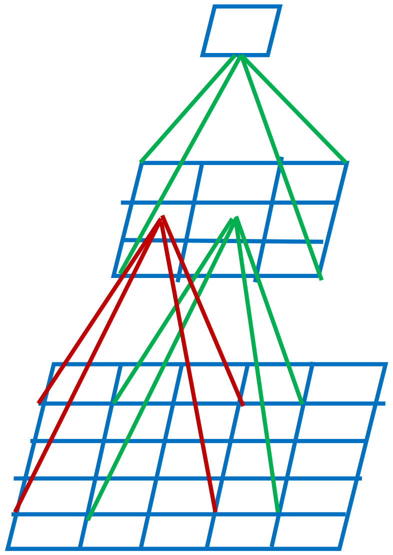
VGG: 3*3 conv

- **19 layers** Deep CNN！ (hard to image before)
  - Resulted in **8.81%** top-5 error in ImageNet
- Only 3*3 kernel Convolution is used



VGG: 3*3 conv

- Why 3*3 Conv is all you need?
  - Stacking 3*3 Conv has large receptive field
    - Stacking 2 = 5*5   stacking 3 = 7*7
  - Stacking 3*3 Conv has less parameters
    - 9*2 = 18 < 25 and 9*3=27< 49

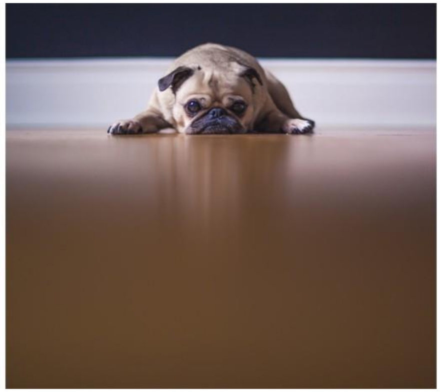


VGG: 3*3 conv

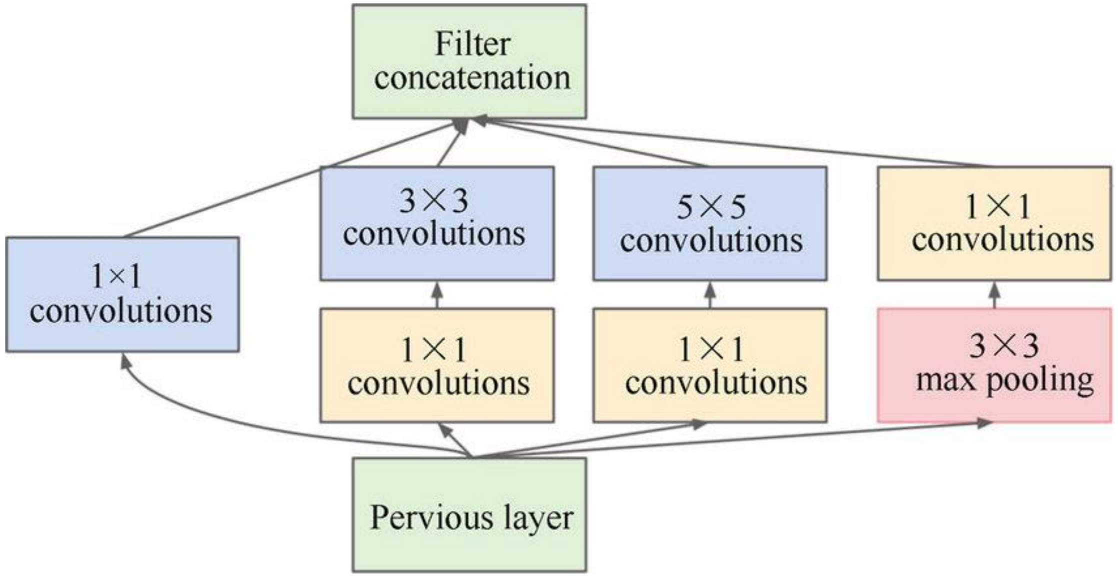
- Xavier Initialization: make the weights have proper scale
  - Too small weights: variance of the input signal makes no difference
  - Too large weights: too sensitive to the small input changes
  - Variance(y=wx) = Variance(x)
  - var(w) = 1/channel#

Inception: multi-scale

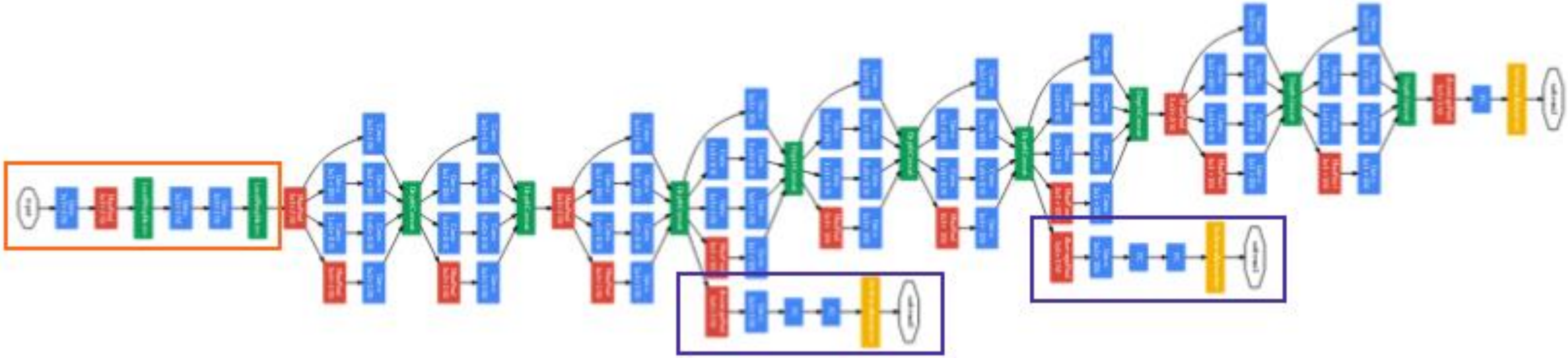
Multi-scale Salient regions



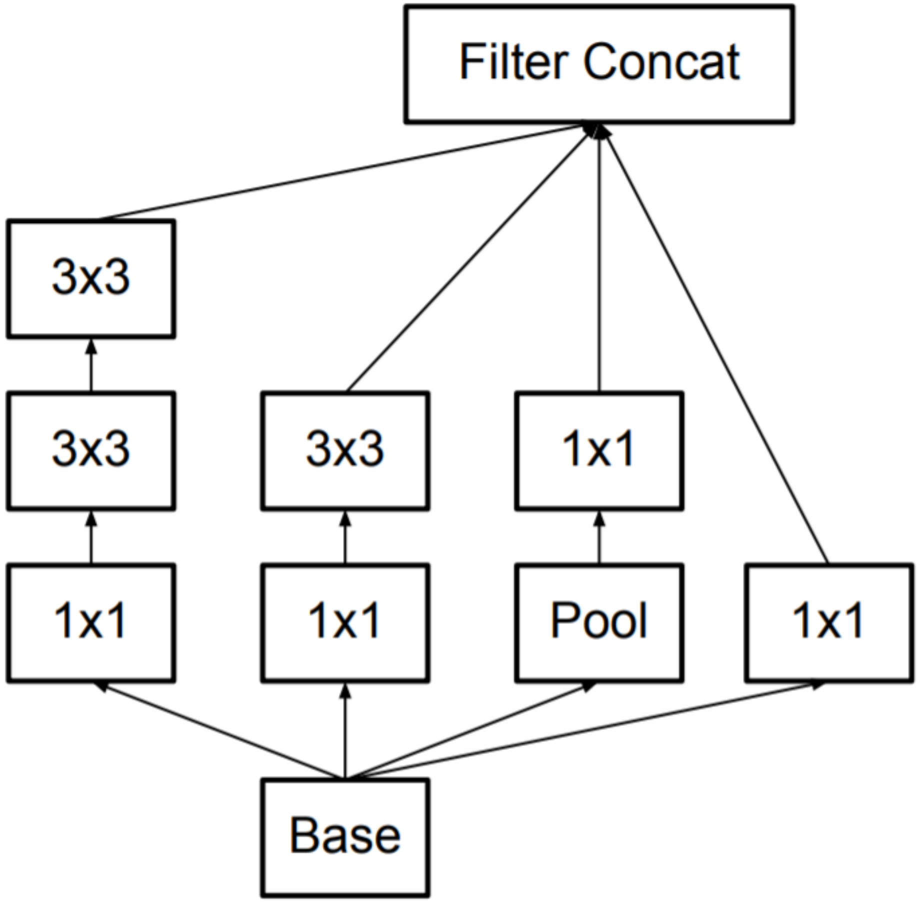
Modeling Multi-scale:
Winner of ImageNet 2014
（Top 5 error 7.89%）

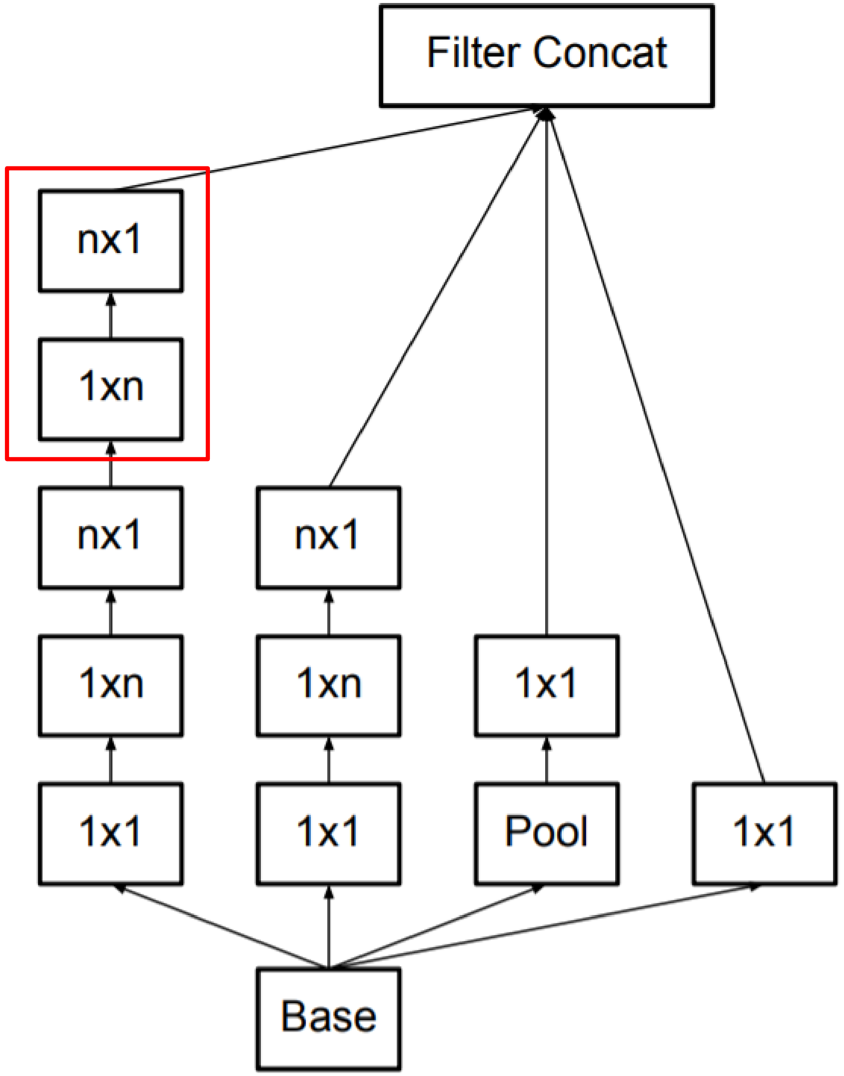


Inception: multi-scale



Inception: multi-scale





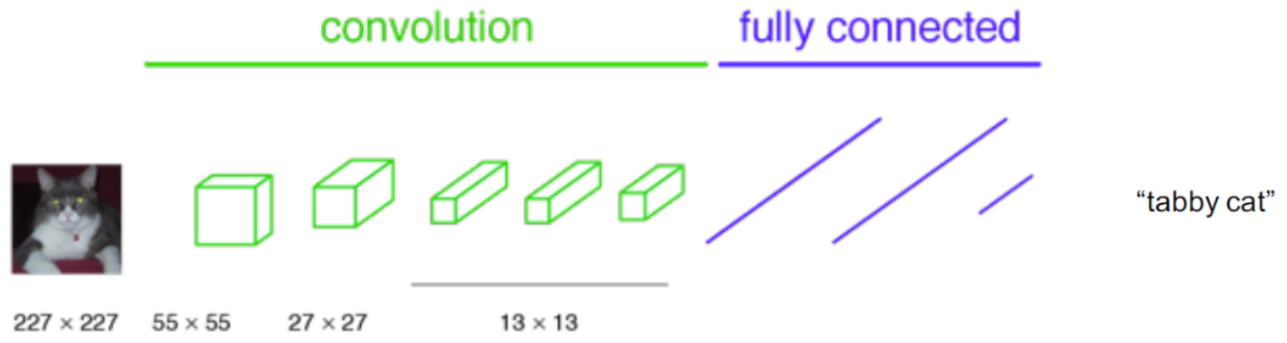
FCN: 1*1 Conv

- Segmentation: Pixel-level Classification

- Naïve Solution: Patch Classification + Sliding window
    - Sliding windows have overlap
    - Repeated computation in the low level feature

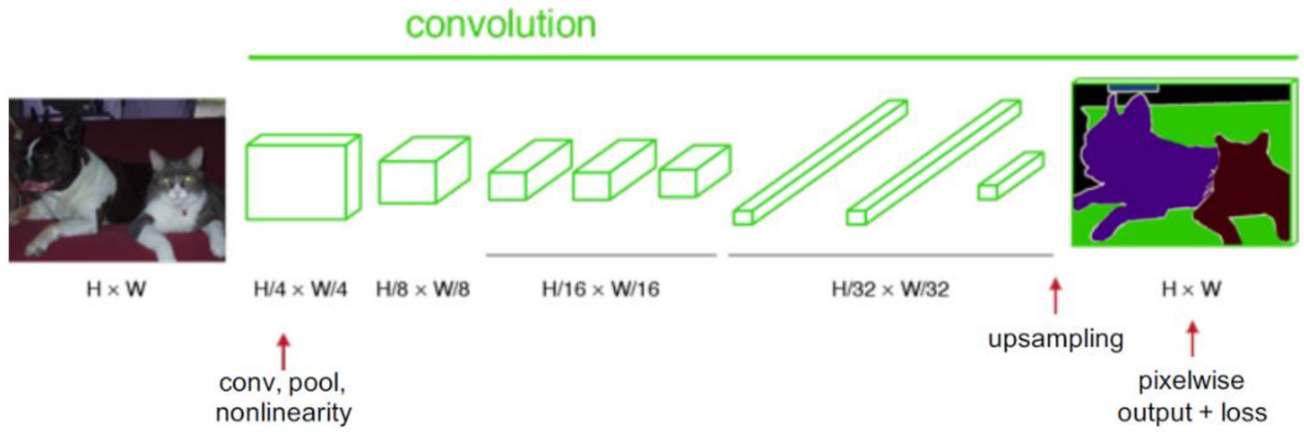
FCN: 1*1 Conv

- Fully Convolutional Network



Classification

Global Average Pooling



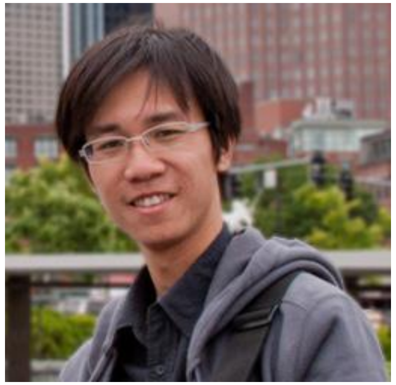
1*1 Conv on every pixel

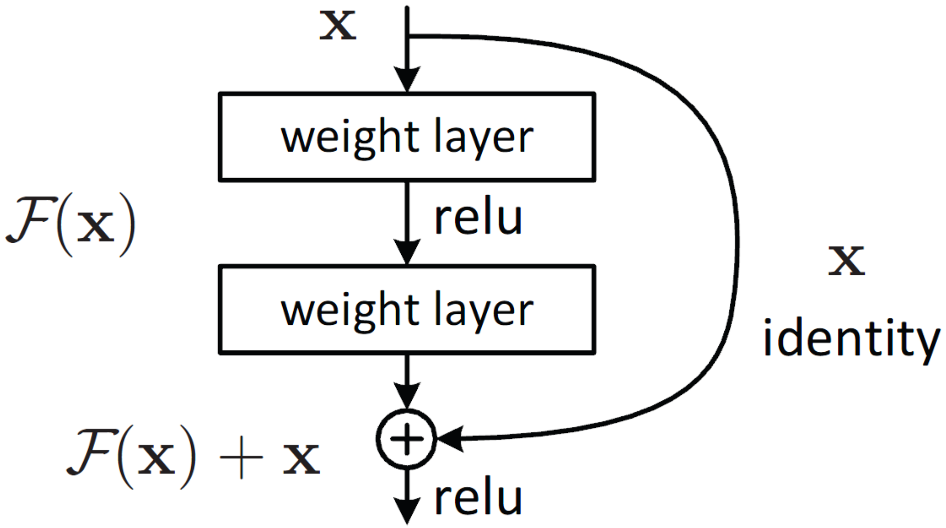
Resnet: Residual Connection

- Extreme Deep Network: 152 layers
- Top 5 errors: **6.34% winner of ImageNet 2015**

- Hard to train a very deep CNN
  - Gradient Vanish
  - Gradient Explosion

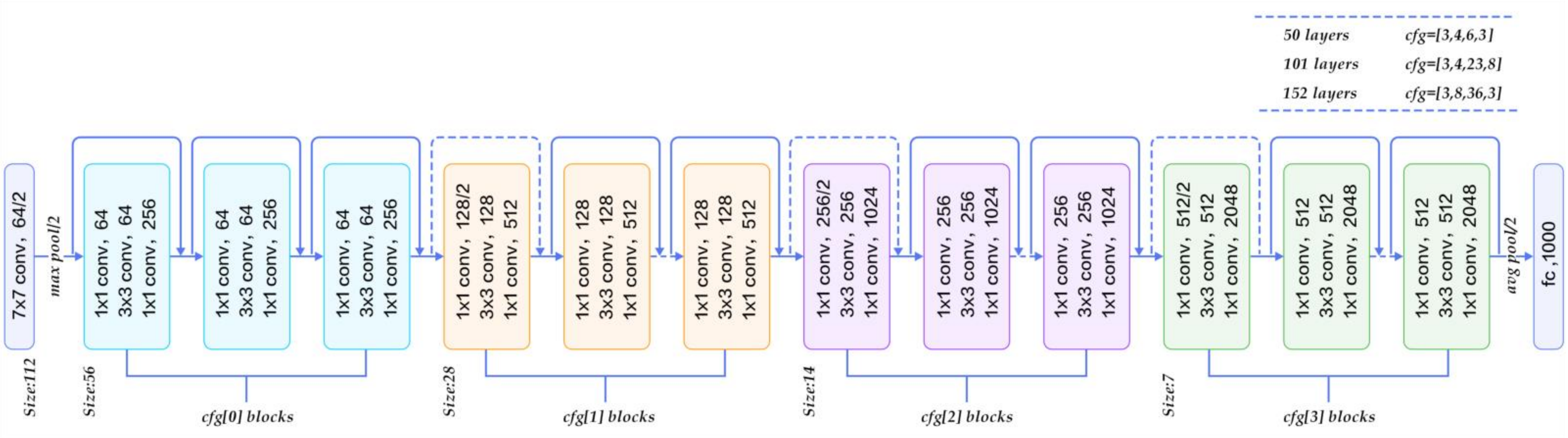
- Solution: Residual Connection!
  - Direct Gradient to low layers
  - Regularization: Learn to skip layers

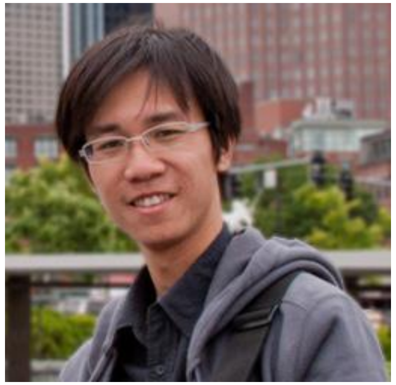




Resnet: Residual Connection

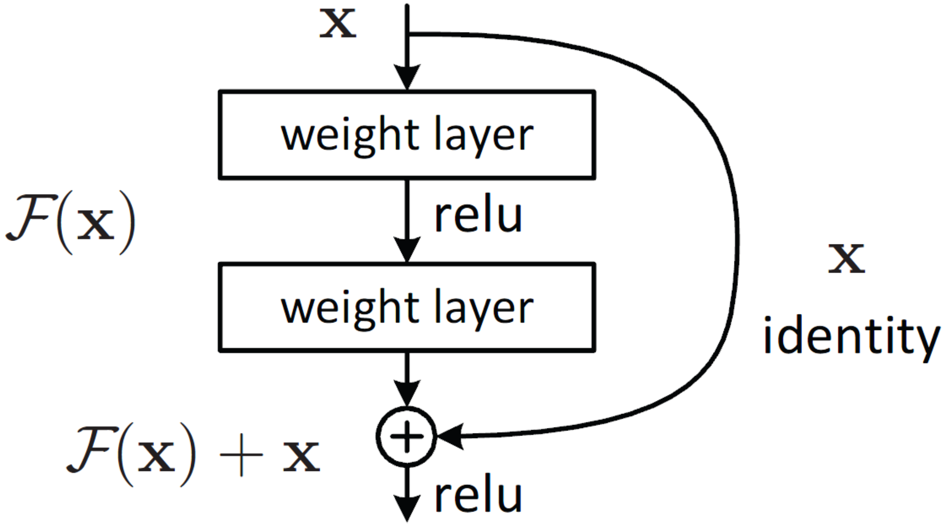
Hong Kong Society of Artificial Intelligence and Robotics
香港人工智能與機器人學會





- Gradient Explosion

- Solution: Residual Connection!
  - Direct Gradient to low layers
  - Regularization: Learn to skip layers

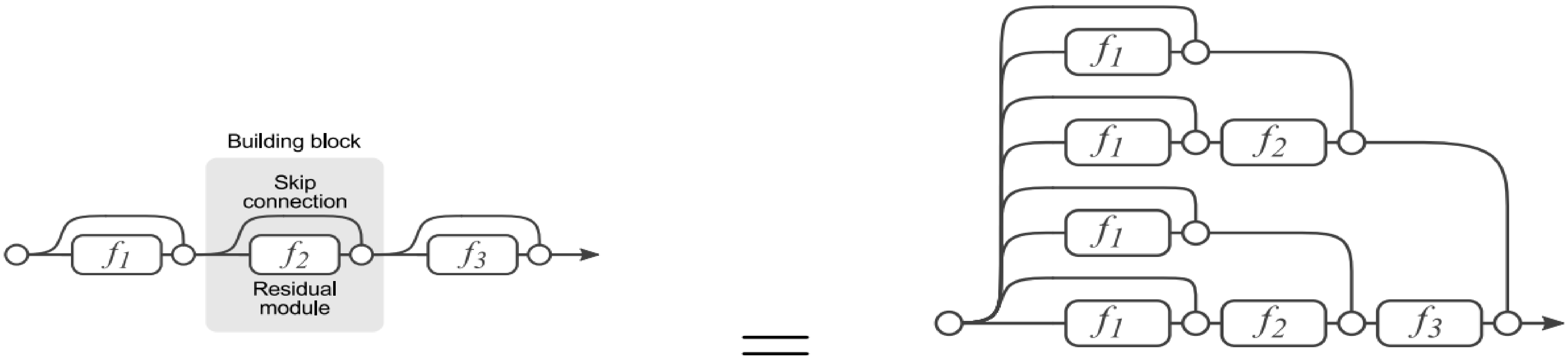


Resnet: Residual Connection

- Boosting: $\mathrm{F}(x) = \sum_{\mathrm{i}} h_i(x)$

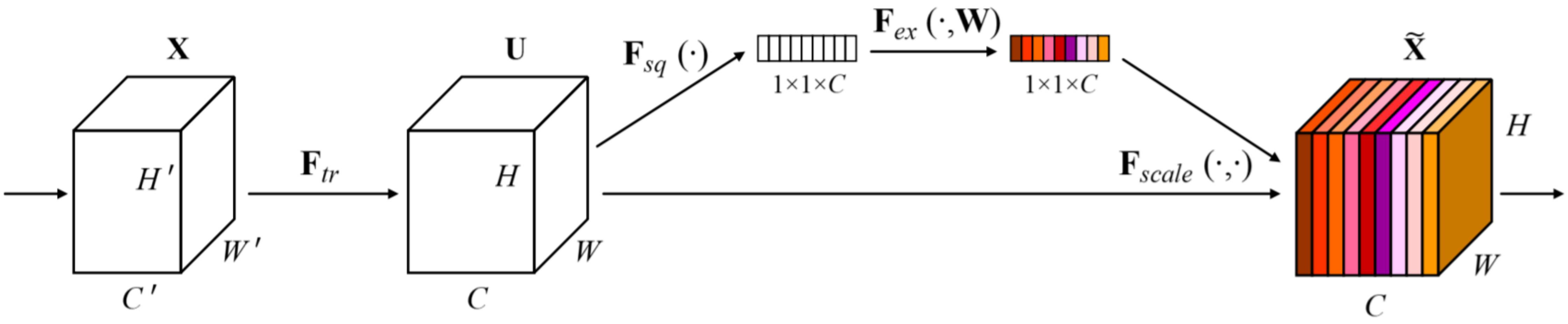
1. Fit a model to the data, $f_0(x) = y$, $F_0(x) = f_0(x)$
2. Fit a model to the residuals, $f_{i+1}(x) = y - F_i(x)$
3. Create a new model, $F_{i+1}(x) = F_i(x) + f_{i+1}(x)$ and repeat step 2

- Residual Connection VS. Boosting

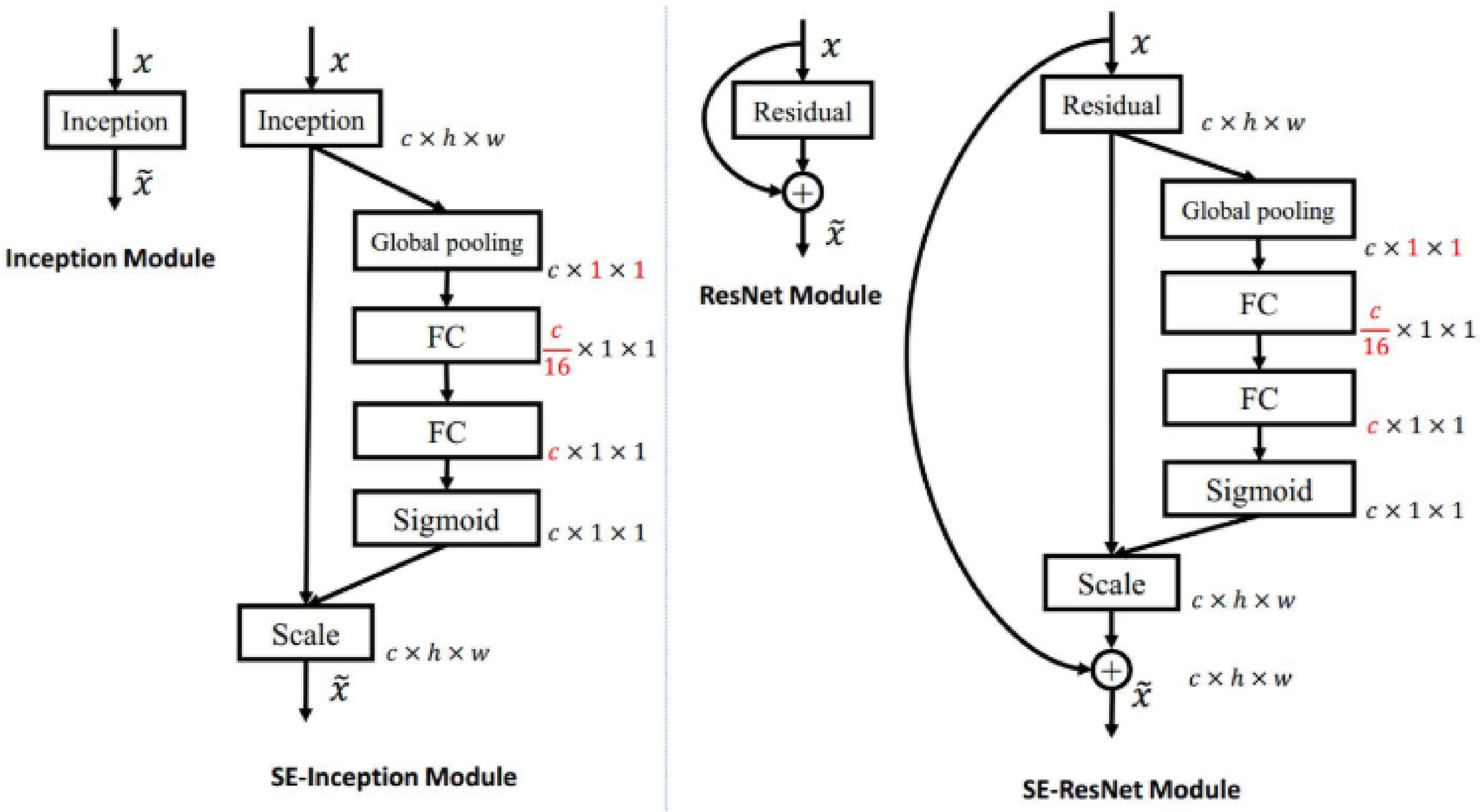


SENet: Independent Channel

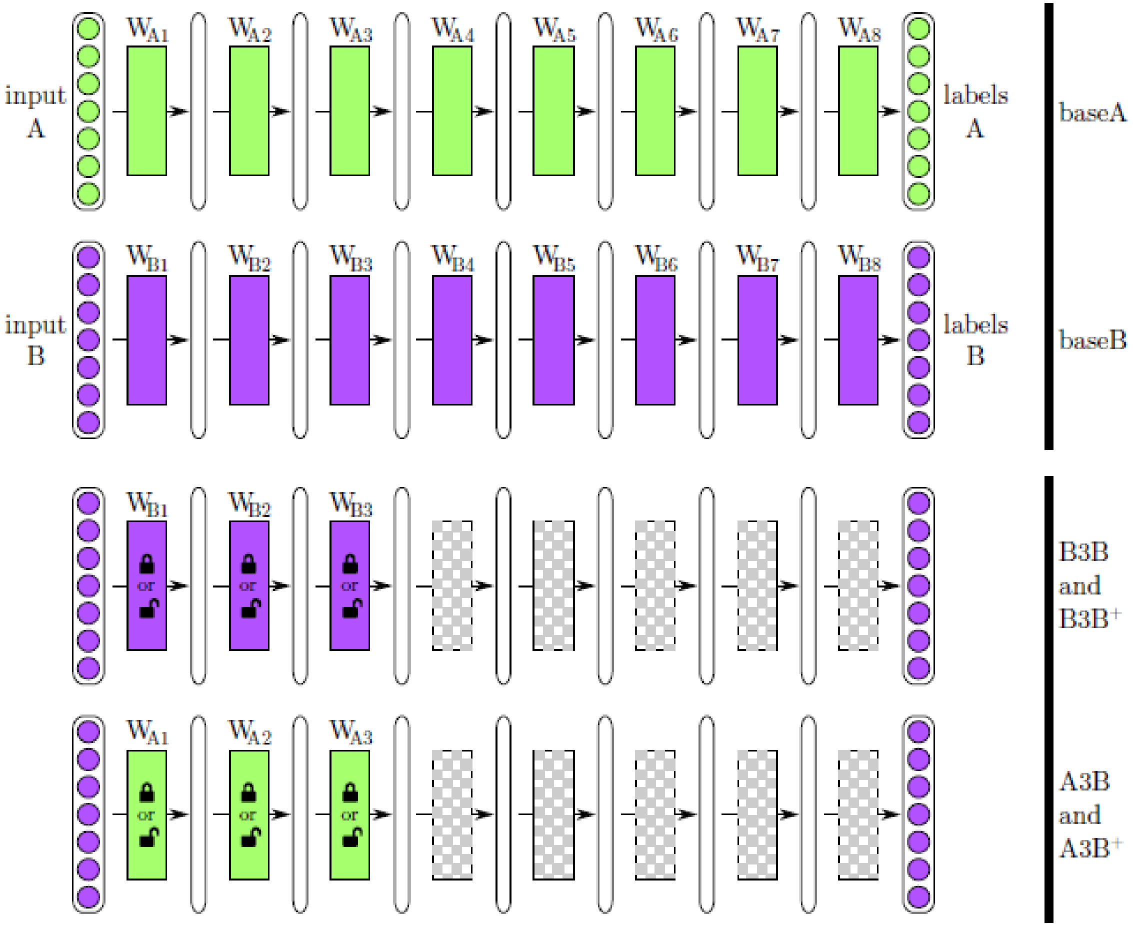
- **2017 ImageNet Winner**: 5.54% top-5 error
- Squeeze: Global Average Pooling to get a global vector with C channels
- Excitation: learn the association of W and output channel-wise weights
-



SENet: Independent Channel



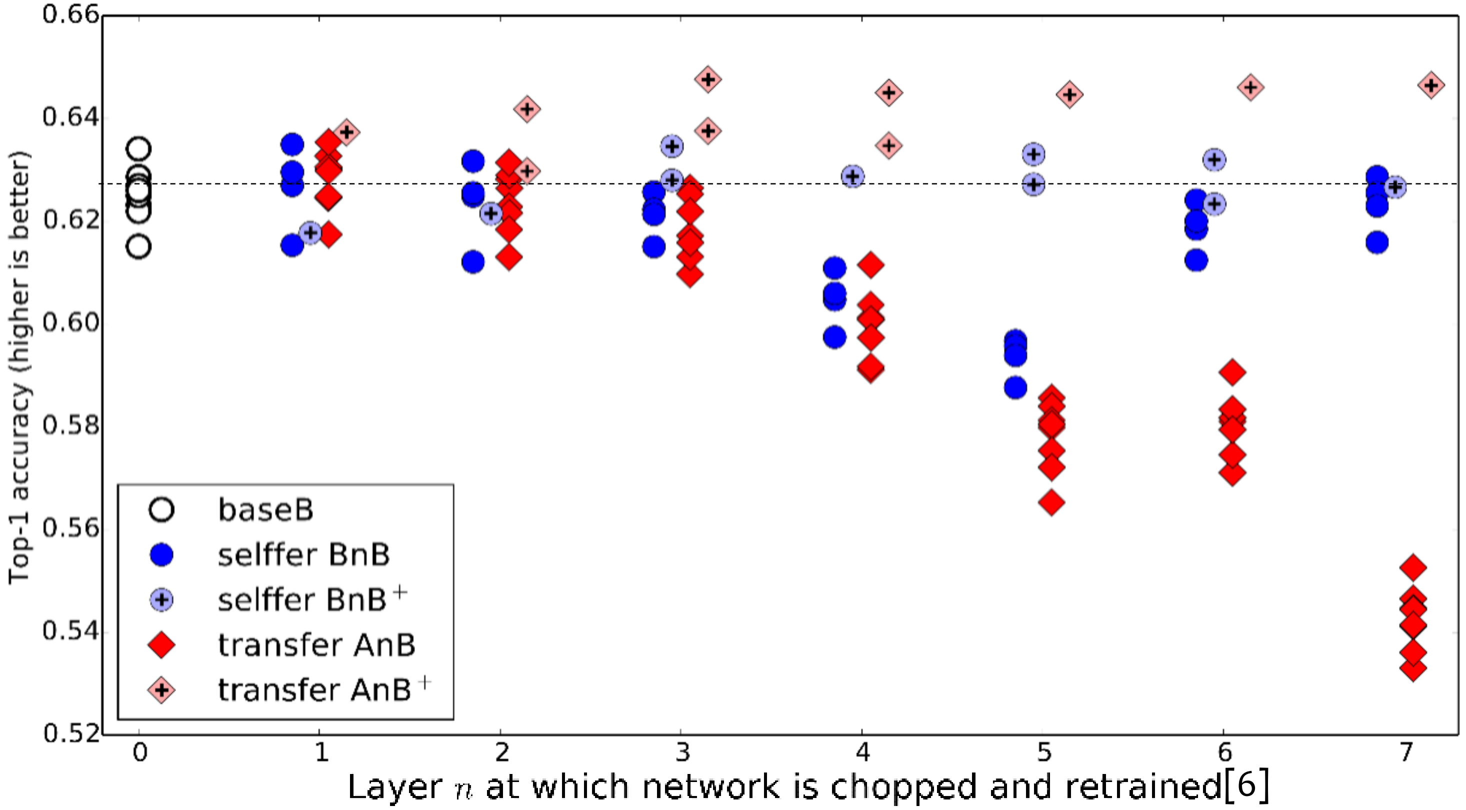
How transferrable CNN is?



How transferrable CNN is?

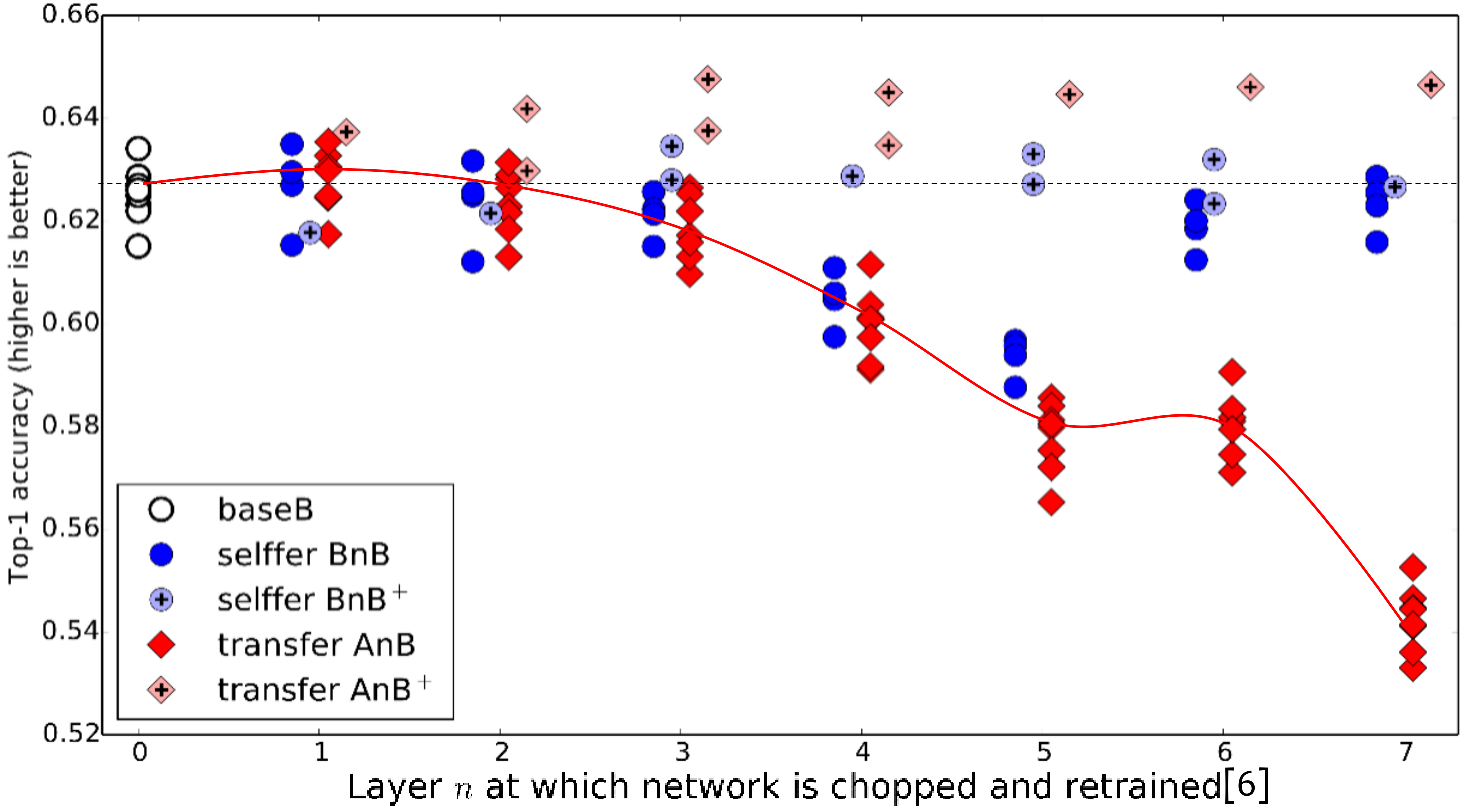
Hong Kong Society of Artificial Intelligence and Robotics 香港人工智能與機器人學會

- Transferability of layer-wise features



How transferrable CNN is?

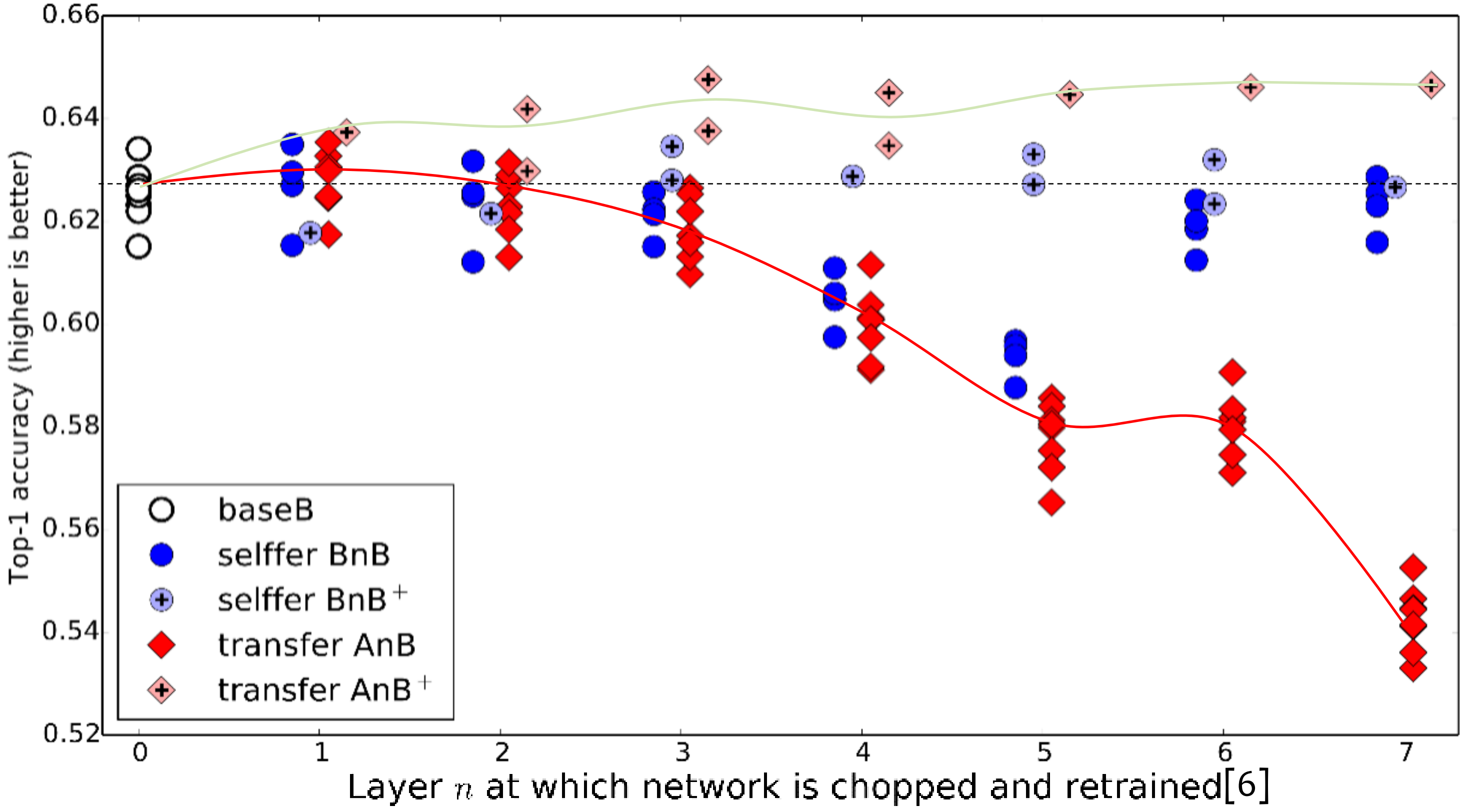
- Transferability of layer-wise features



Conclusion 1: lower layer features are more general and transferrable, and higher layer features are more specific and non-transferrable.

How transferrable CNN is?

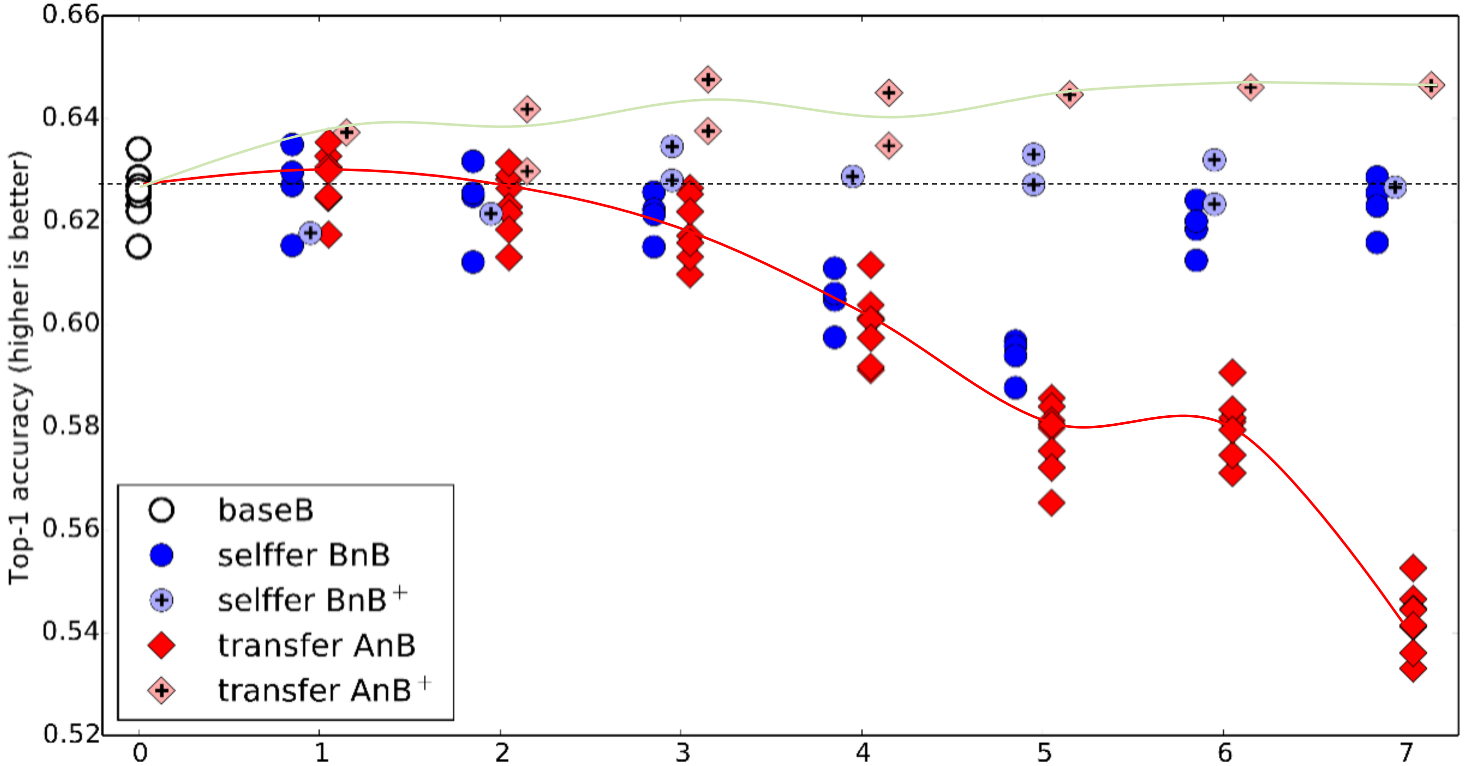
- Transferability of layer-wise features



Conclusion 2: transferring features + fine-tuning always improve generalization.

How transferrable CNN is?

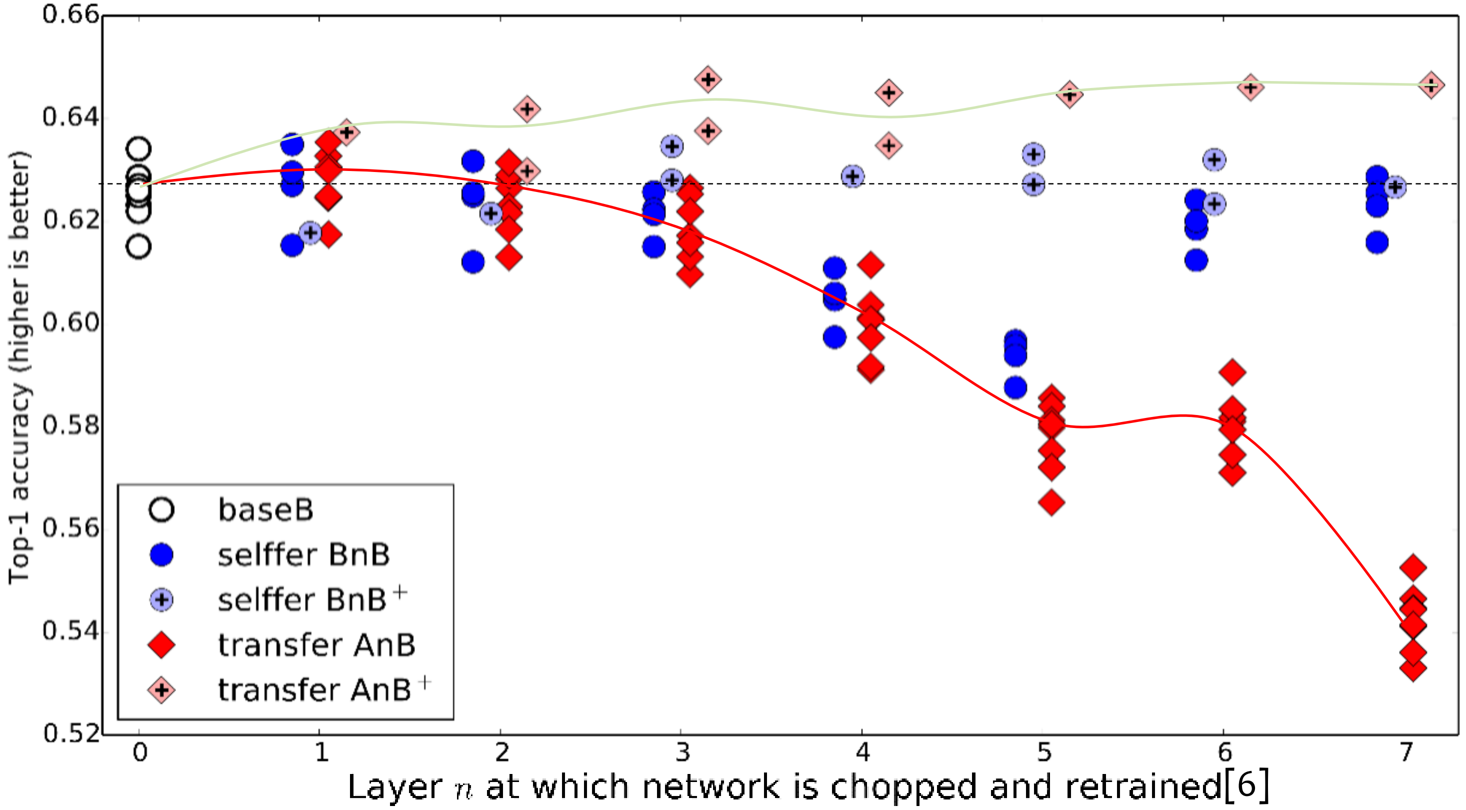
- Transferability of layer-wise features



Layer n at which network is chopped and retrained[6]

How transferrable CNN is?

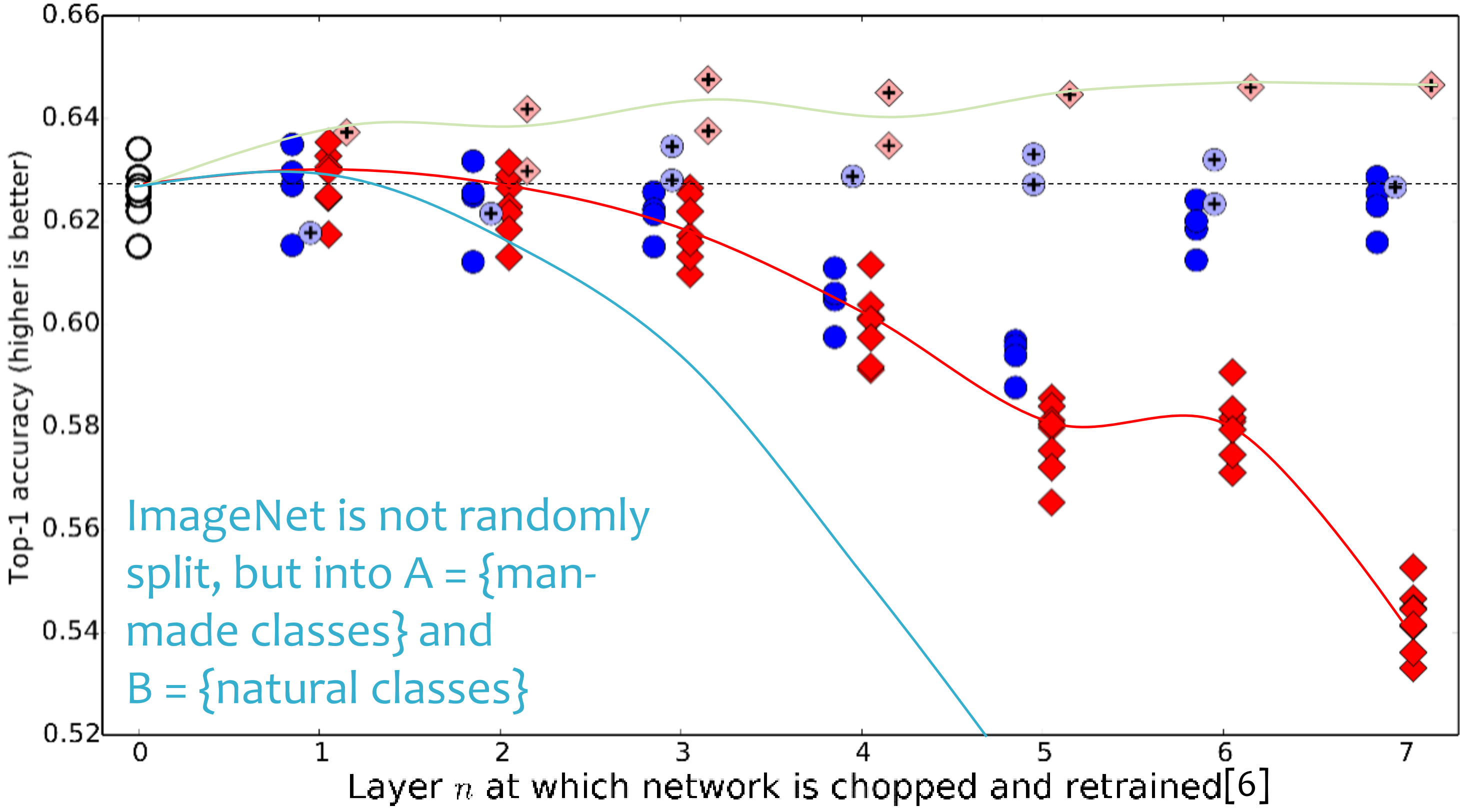
- Transferability of layer-wise features



What happens if the source and target domain are very dissimilar?

How transferrable CNN is?

- Transferability of layer-wise features



What happens if the source and target domain are very dissimilar?

More Things

- CNN for Frequency: Graph CNN, OctoveCNN
- Regression: Object Detection(RCNN series), Counting
- 3D CNN: time-series, CT, MRI …
- Generative Model: GAN, VAE …
- Compact Model/Model Compression
- Automatic Network Architecture Searching
- …

Take-Home Message

- Deep Learning: **Hierarchical Representation Learning**
- Convolution:
  - All linear operation are Conv: 3*3 1*1 and dense
  - spatial localization regularized version of the Dense
- Regularization: less maybe more!
  - Less parameters
  - Normalized range
  - Proper Architecture: Multi-scale, Residual, SE etc.
- It’s a long journey, AI doesn’t happen overnight
  - Alexnet:2012 - LeNet: 1998 = 14 years!

- Next Workshop: Sequence Modeling
  - Natural Language Processing
  - Time Series

Jürgen Schmidhuber

Have Insights and Have Fun!