Transfer learning – Next big thing

PRESENTED BY -
Aayush Agrawal

Aayush Agrawal

Education -

- MS in Business Analytics, Carlson School of Management, University of Minnesota, 2017
- B.Tech in Electrical Engineering, Malaviya National Institute of Technology, India, 2013

Experience –

- >4 years in Data science, Currently working as Data scientist for Land O’ Lakes, Inc.
- Moderator and rank 3rd at https://www.analyticsvidhya.com/
- Kaggle Expert - https://www.kaggle.com/aayushmnit

Reach me out at (@aayushmnit) –

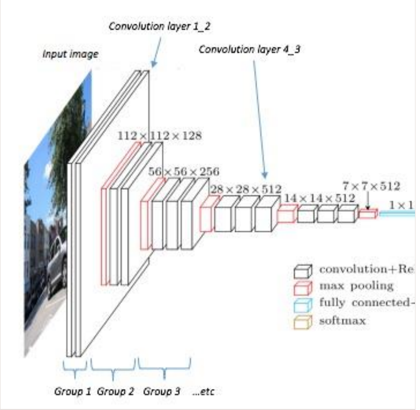
Email - aayushmnit@gmail.com

LinkedIn - https://www.linkedin.com/in/aayushmnit/

Github - https://github.com/aayushmnit

Agenda

Introduction to CNN's



What/WHY

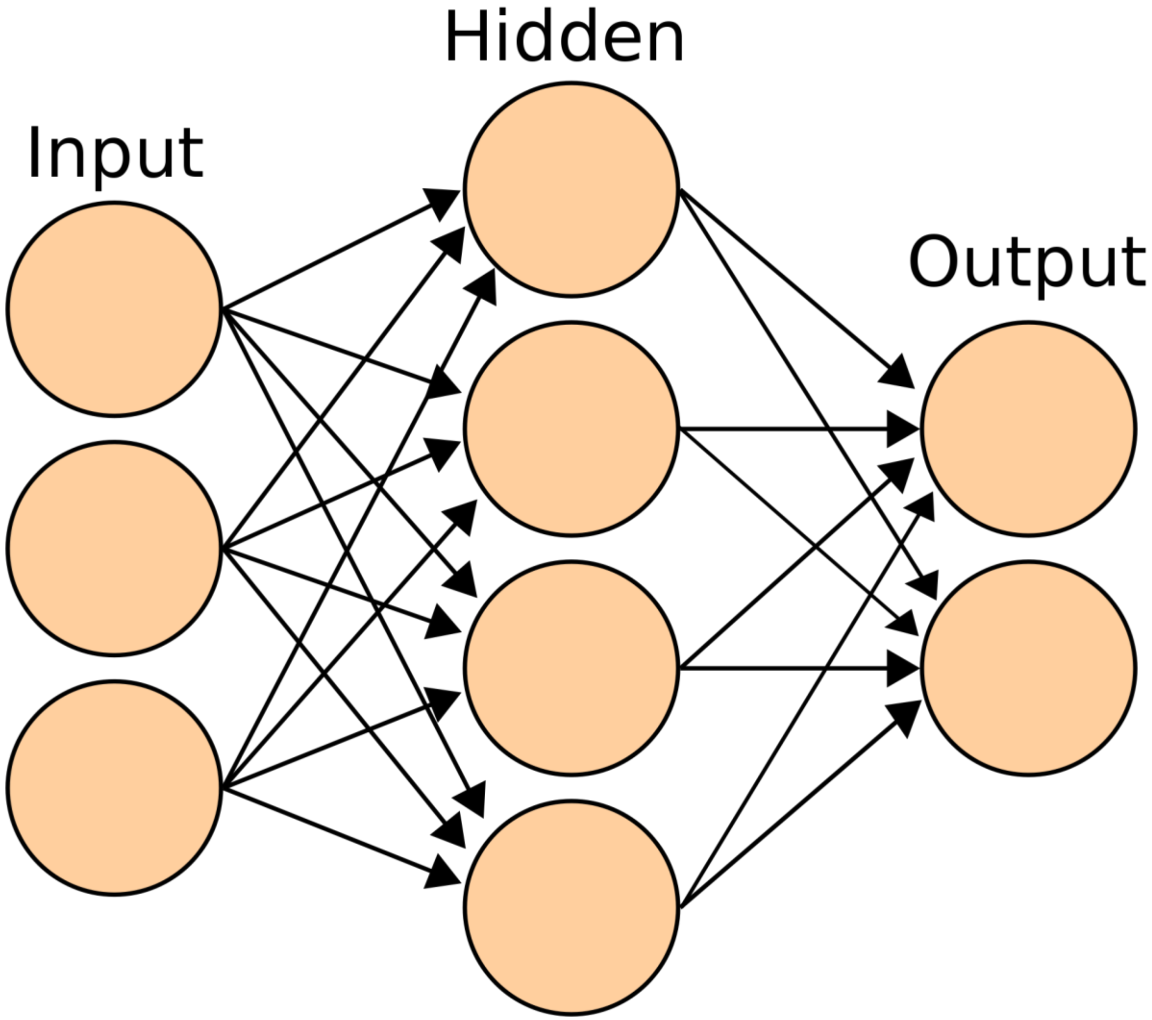
Transfer learning?

How to do

Transfer learning

Examples

Traditional NN don't scale well to full images

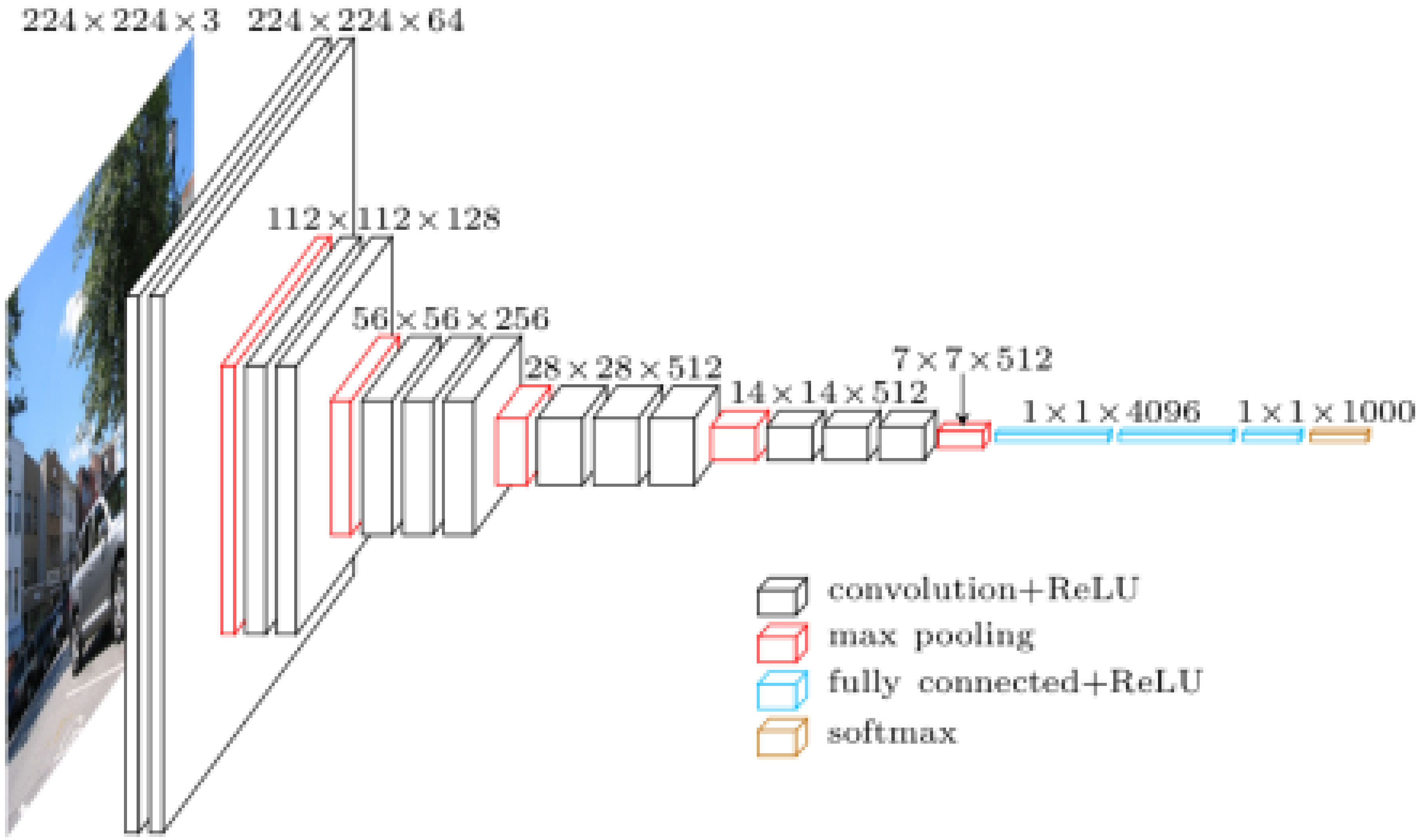


- Consider an image of size 32*32*3 (CIFAR-10 images), each neuron have 32*32*3 = 3072 weights
- A low resolution image these days are around 200*200*3 while leads to 120,000 weights
- Also we need more than one neuron obviously
- We also need more than one layer

Convolution NN takes advantage that input consists images

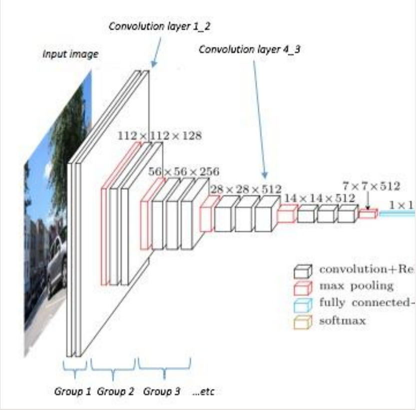
- Typically a ConvNets will consist of three layers
  - Convolutional Layers
  - Pooling layer
  - Fully – Connected layer
- Basic architecture - **[INPUT - CONV - RELU - POOL - FC]**
- **Convolution** layer will compute the output of neurons that are connected to local regions in the input, each computing a dot product between their weights and a small region they are connected to in the input volume (https://distill.pub/2016/deconv-checkerboard/)
- **Relu** layer will apply an elementwise activation function, such as the max(0,x) thresholding at zero.
- **Pool** layer will perform a downsampling operation along the spatial dimensions (width, height)
- **Fully connected** layer will compute the classes. Each neuron in this layer is connected to all the inputs

A basic VGG16 architecture



Agenda

Introduction to CNN’s



What/WHY

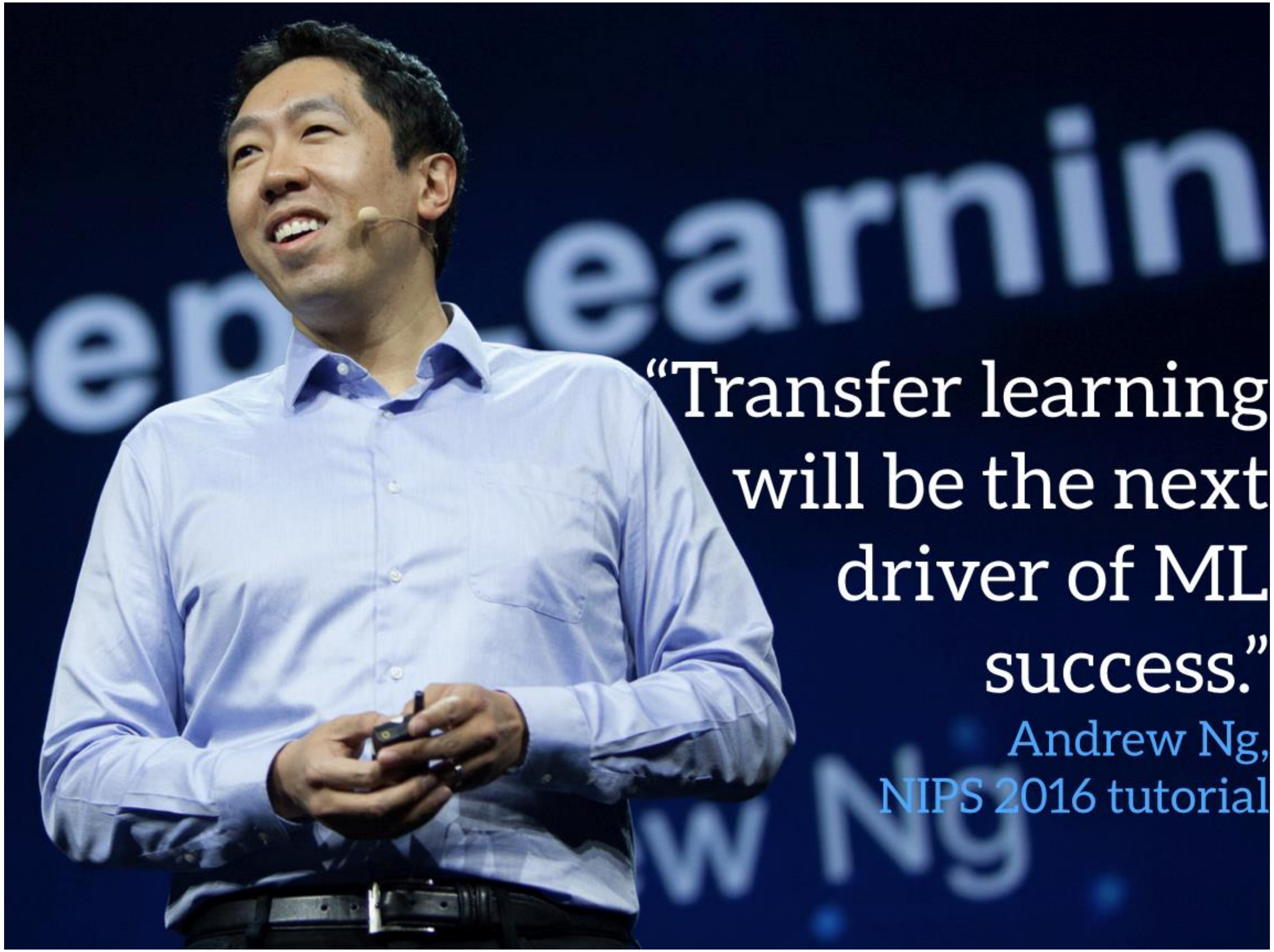
Transfer learning?

How to do

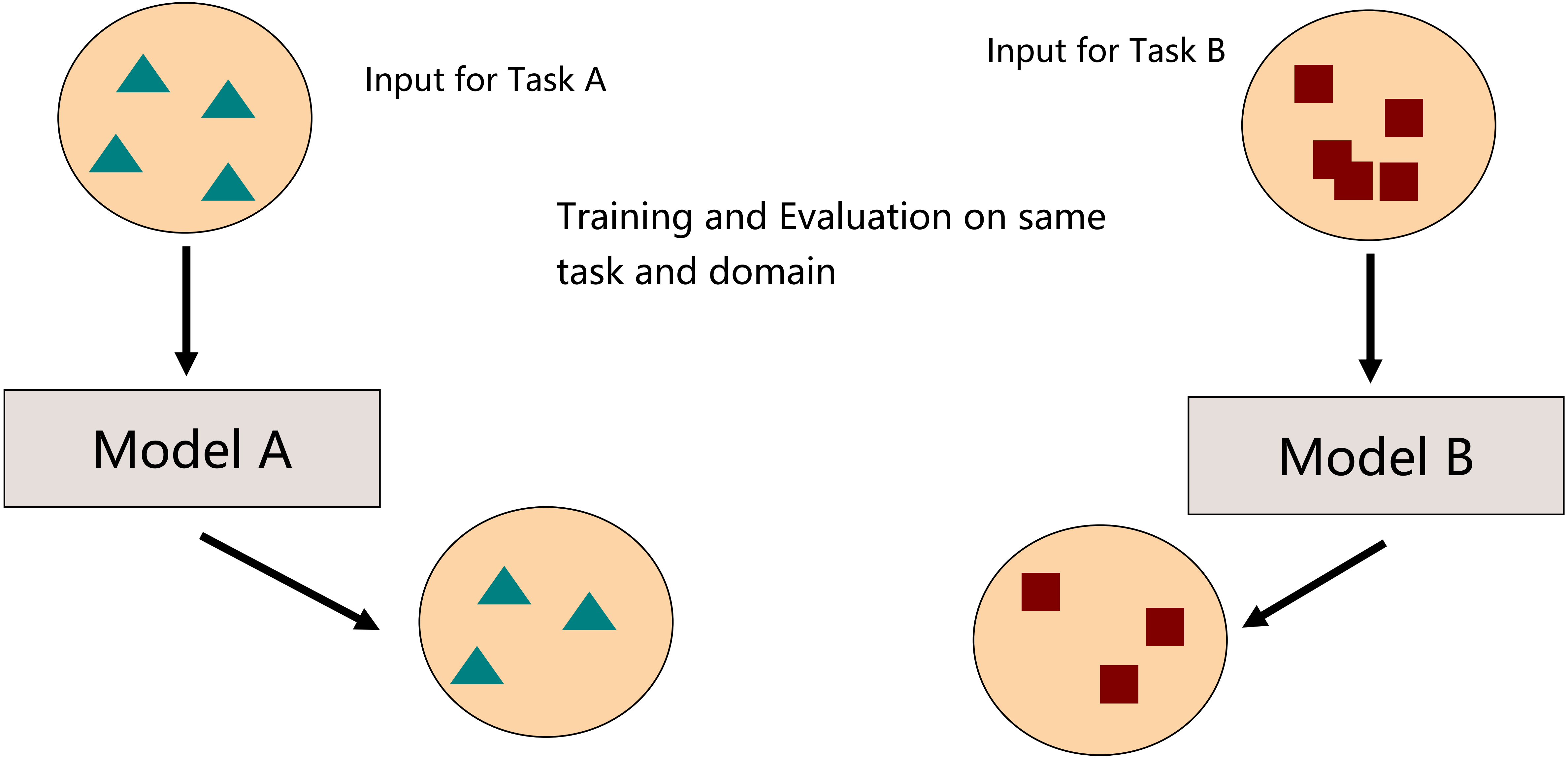
Transfer learning

Examples

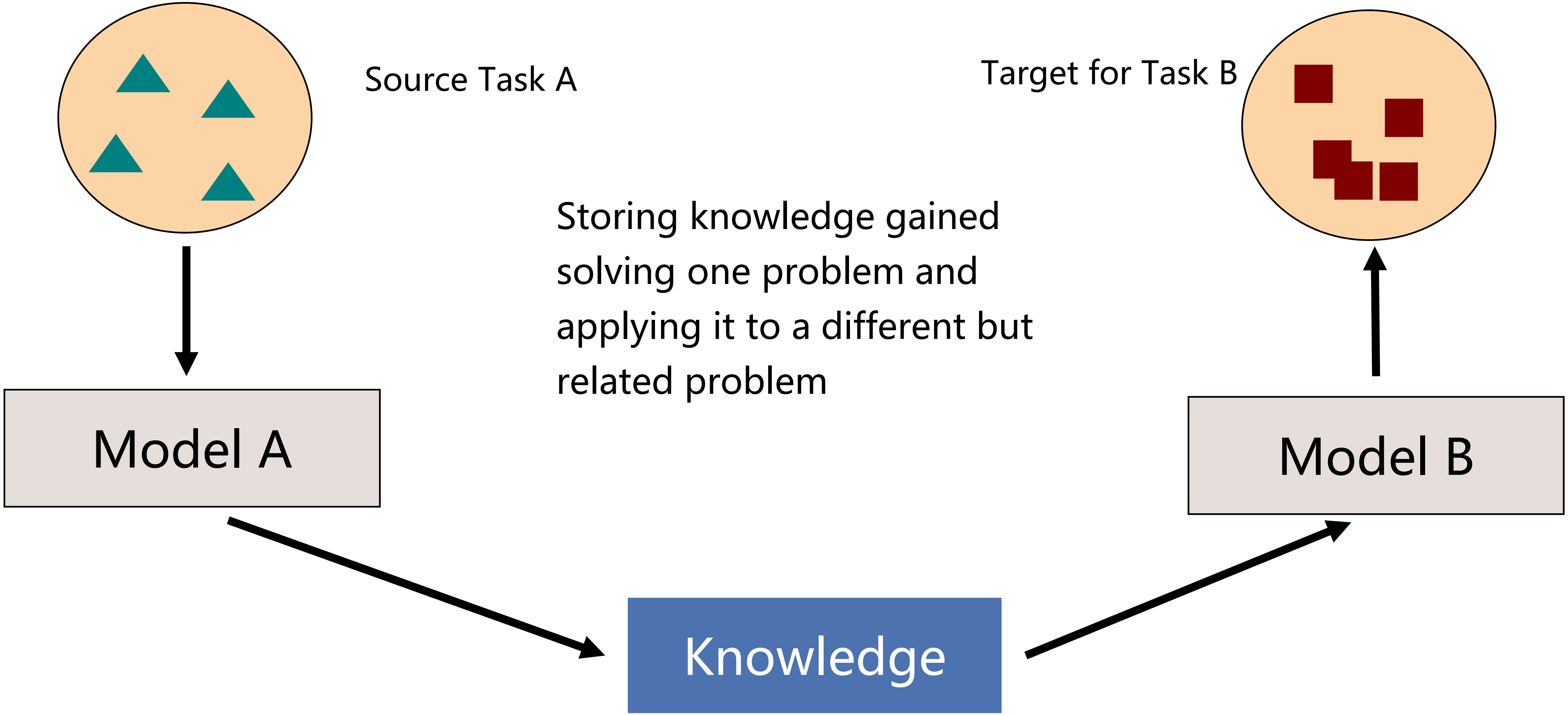
Because Andrew NG said so ….



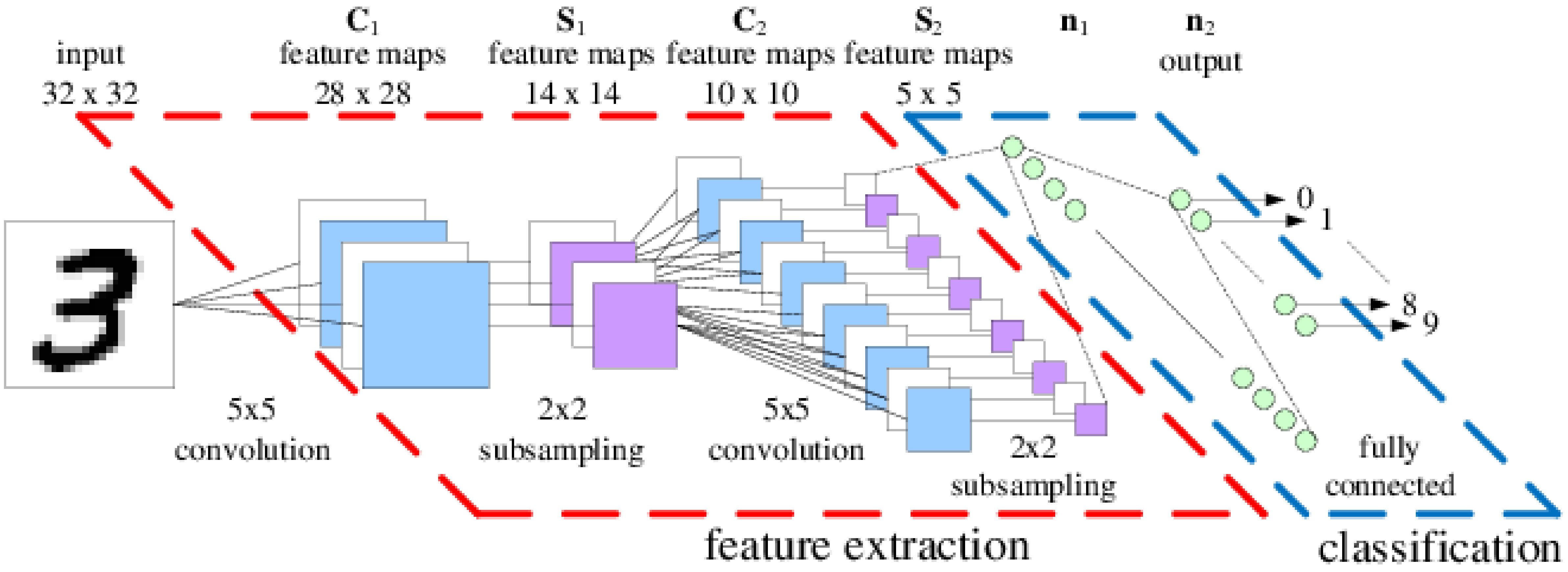
Classic machine learning requires retraining from start



Transfer learning allows to transfer knowledge from one task to another



A large part of ConvNet is transferable



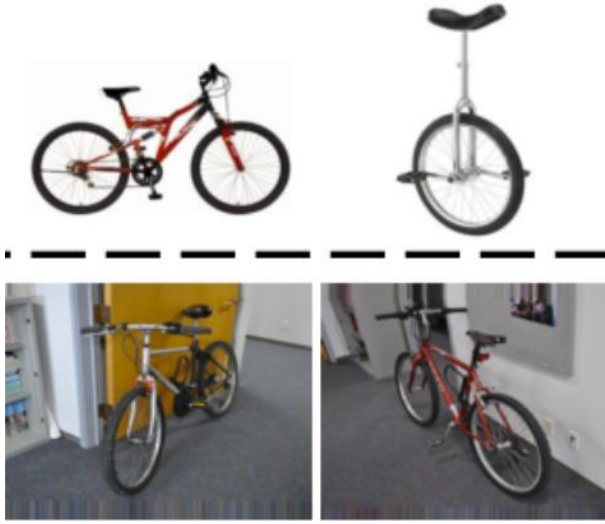
Generic and can be transferred

Task specific

Transfer learning allows model to adapt for unseen scenarios

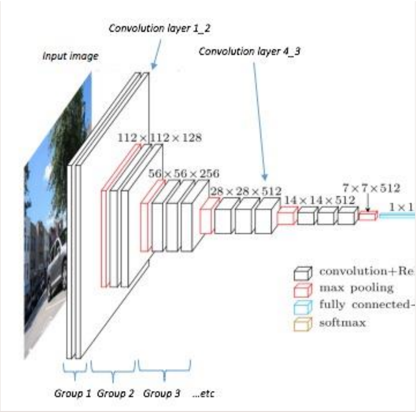
- We don’t need extensive amount of data for a specific real scenario to train a deep learning model

- Major applications of transfer learning
  - **Learning from simulations**
    - Self driving cars can learn from data collected from driving car in GTA5
    - Robotics simulation can ease the data collection process from a real robot
  - **Adapting to new domains**
    - General image classification can be used to train on new domains
    - Speech models like Glove, Word2vec can be used to adapt to new domain specific text
    - Voice recognition system built adapting to kids voice



Agenda

Introduction to CNN's



What/WHY

Transfer learning?

How to do

Transfer learning

Examples

Transfer learning in ConvNet

Three major transfer learning scenarios look as follows:

- **ConvNet as feature extractor**
  - Take a ConvNet pretrained on dataset like ImageNet
  - Remove the last fully connected layer
  - Treat the last as a feature extractor for the new dataset
  - Run a linear/tree based classifier on new features

- **Fine Tuning the ConvNet**
  - Replace the classifier of a Convnet saved on Imagenet
  - Fine tune model weights of all the layers present in the model using new dataset
    - Tuning the last few layers give maximum increase in accuracy

- **Using a pretrained model**
  - This involves using pretrained weights of someone else to finetune the existing model

When and how to finetune?

Scenario 1: *New dataset is small and similar to original dataset*
Recommendation: Use a Convnet as feature extractor and train a linear model on top of it, because of overfitting issues

Scenario 2: *New dataset is large and similar to the original dataset*
Recommendation: Fine tuning the ConvNet by adding new fully connected layer would be best as we can prevent the overfitting issues because of large data size.

Scenario 3: *New dataset is small but very different from the original dataset.*
Recommendation: Don't use a pretrained model and go for linear models only

Scenario 4: *New dataset is large and very different from the original dataset*
Recommendation: Train a Convnet from scratch. It is very often still beneficial to initialize with weights from a pretrained model. Fine tuning the entire network is best case scenario.

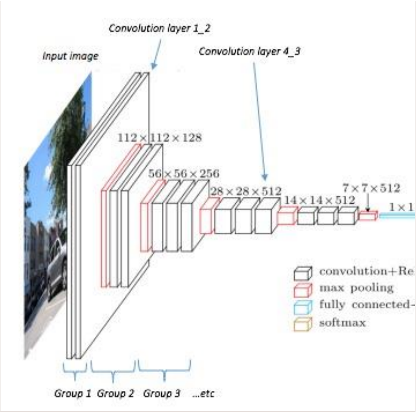
Practical advice/drawback using pretrained models

- *Constraints from pretrained models*
  - *Restricted by the architecture used for training pretrained model*
  - *You can't arbitrarily take out Convolutional layers from the model*

- *Learning Rates*
  - *Always start with a small learning rate for ConvNet weights that are being fine tuned*
  - *We expect ConvNet weights to be good enough to be used, so we don't wish to distort them too quickly and too much*

Agenda

Introduction to CNN's



What/WHY

Transfer learning?

How to do

Transfer learning

Examples

Deep learning path Suggestion

Data science path - https://www.analyticsvidhya.com/blog/2017/01/the-most-comprehensive-data-science-learning-plan-for-2017/

My Deep learning track –

1) Machine learning by Andre NG(his first course and the most popular course in MOOC history) -> https://www.coursera.org/learn/machine-learning (Low difficulty)

2) Deep learning by Google on udacity - https://www.udacity.com/course/deep-learning--ud730 (Hard)

3) Practical deep learning for Coders by Jeremy Howard (Former Kaggle #1) - http://course.fast.ai/ (Medium/Hard)

4) A book on deep learning (Goodfellow) - http://www.deeplearningbook.org/ (If you need to understand deep math)

5) Andrew NGs deep learning track - https://www.coursera.org/specializations/deep-learning (easy/medium)

6) Just some collection of good blogs - http://colah.github.io/

References for transfer learning used in presentation

- http://ruder.io/transfer-learning/
- http://cs231n.github.io/transfer-learning/
- http://course.fast.ai/
- https://github.com/fastai/courses

Thank You!