Explore and have fun with TensorFlow

Poo Kuan Hoong

Google I/O Extended 2017 brings out the best Google technologies all the way from Mountain View to Kuala Lumpur.

22 July 2017

Google Malaysia, Kuala Lumpur

#io17extended

About me

Poo Kuan Hoong, http://www.linkedin.com/in/kuanhoong

ADAX

- Senior Data Scientist

nielsen

- Senior Manager Data Science

MMU
MULTIMEDIA UNIVERSITY

- Senior Lecturer
- Chairperson Data Science Institute

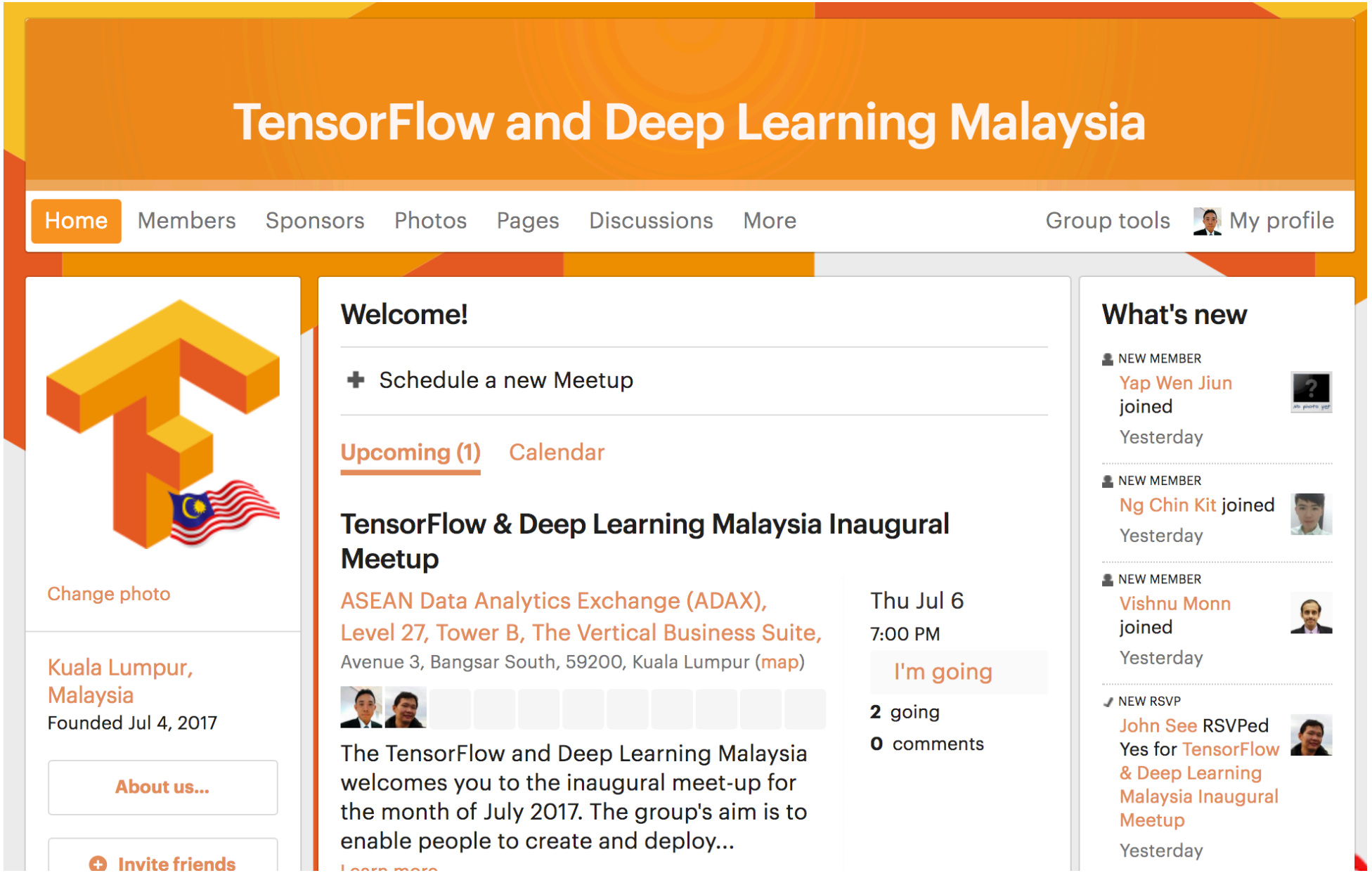
- Founder R User Group & TensorFlow User Group
- Speaker/Trainer

TensorFlow & Deep Learning Malaysia Group

The TensorFlow & Deep Learning Malaysia group's aims are:

- To enable people to create and deploy their own Deep Learning models built using primarily TensorFlow or other Deep Learning libraries.
- To build the key skill sets for this group from the combination of both beginner and intermediate models as well as advancing to the next level
- A knowledge sharing and presentations platform in relation to the cutting edge deep learning research papers and techniques.

https://www.facebook.com/groups/TensorFlowMY/



https://www.meetup.com/tensorflow-deep-learning-malaysia

Malaysia R User Group - (myRUG)

Home | Members | Sponsors | Photos | Pages | Discussions | More | Group tools | My profile

Change photo

Kuala Lumpur, Malaysia
Founded Jun 5, 2016

About us...

Welcome to Malaysia R User Group (myRUG)

+ Schedule a new Meetup

Upcoming | Past | Calendar

There are no upcoming Meetups

You can schedule one!

Schedule a Meetup

Recent Meetups

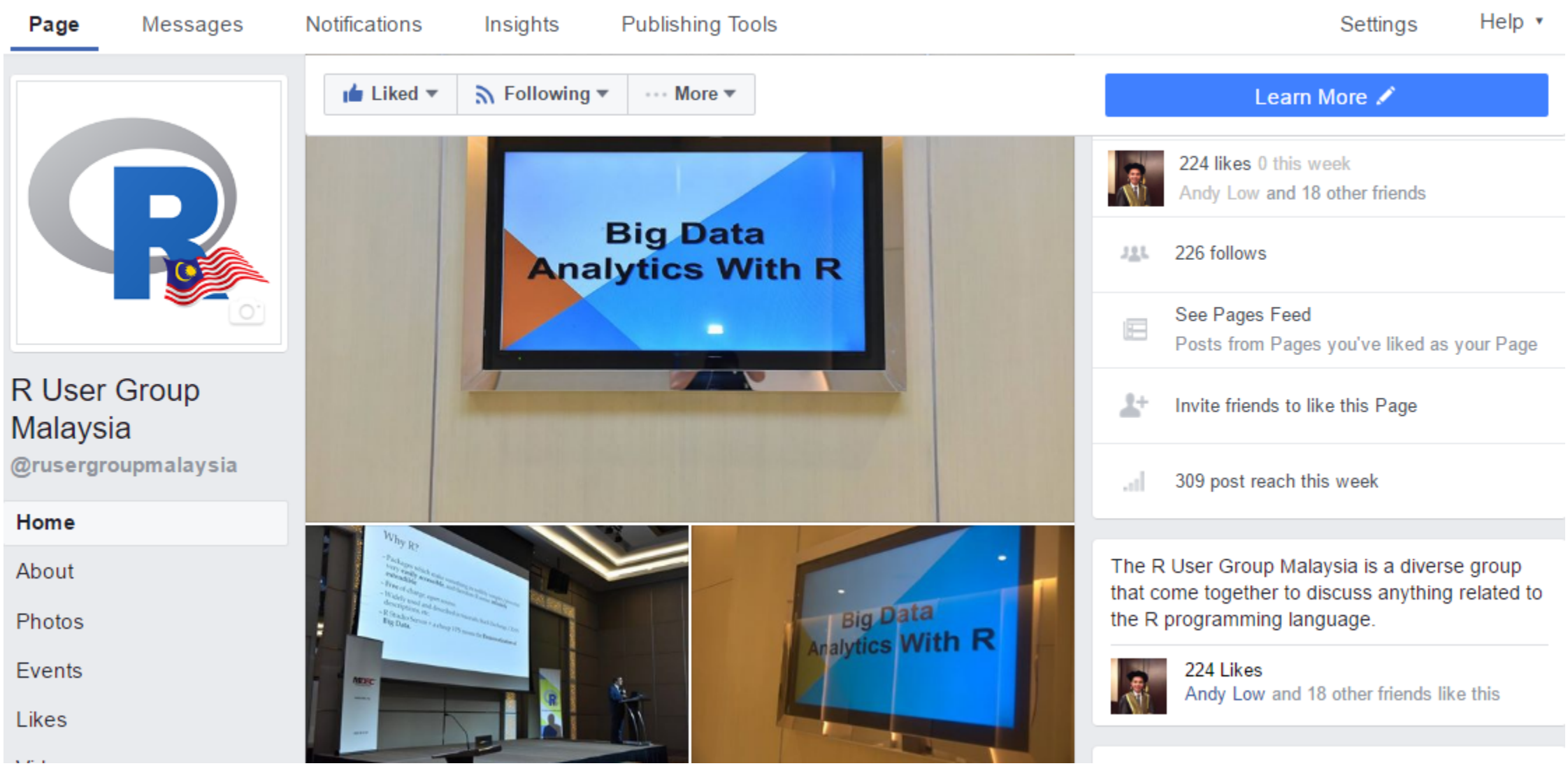
Oct 20, 2016 · 7:00 PM

Rate this Meetup

What's new

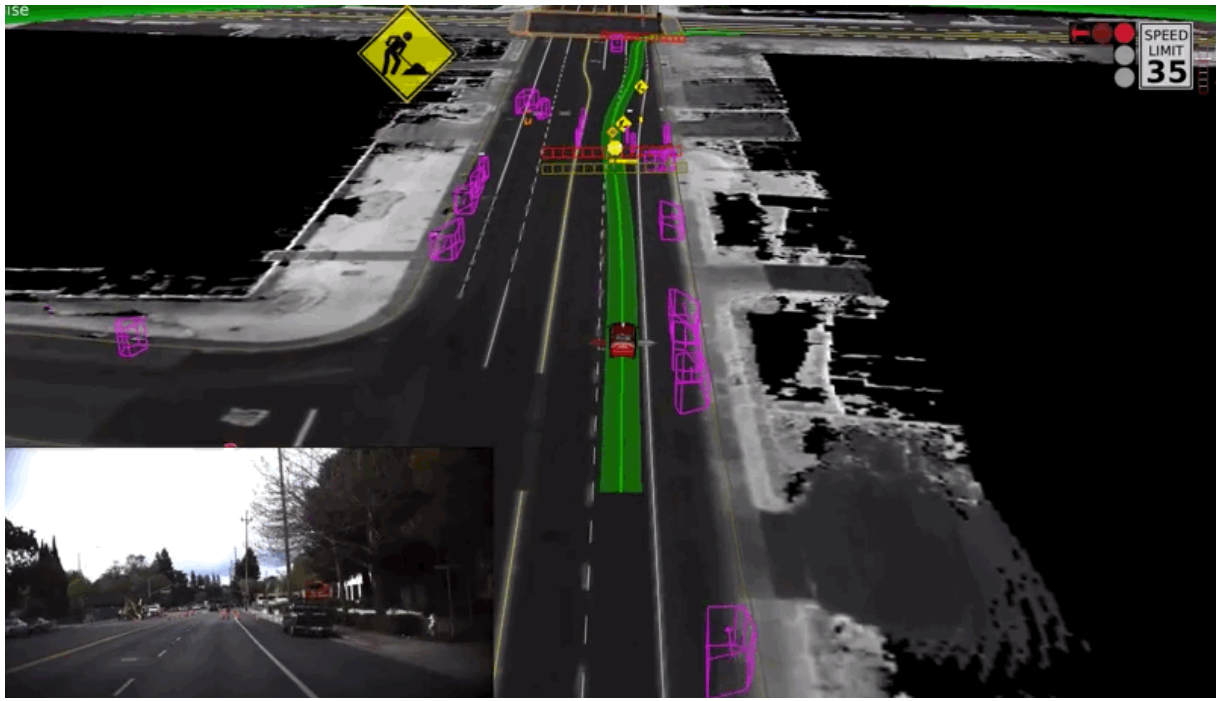
https://www.meetup.com/MY-RUserGroup/

#io17extended



https://www.facebook.com/rusergroupmalaysia/

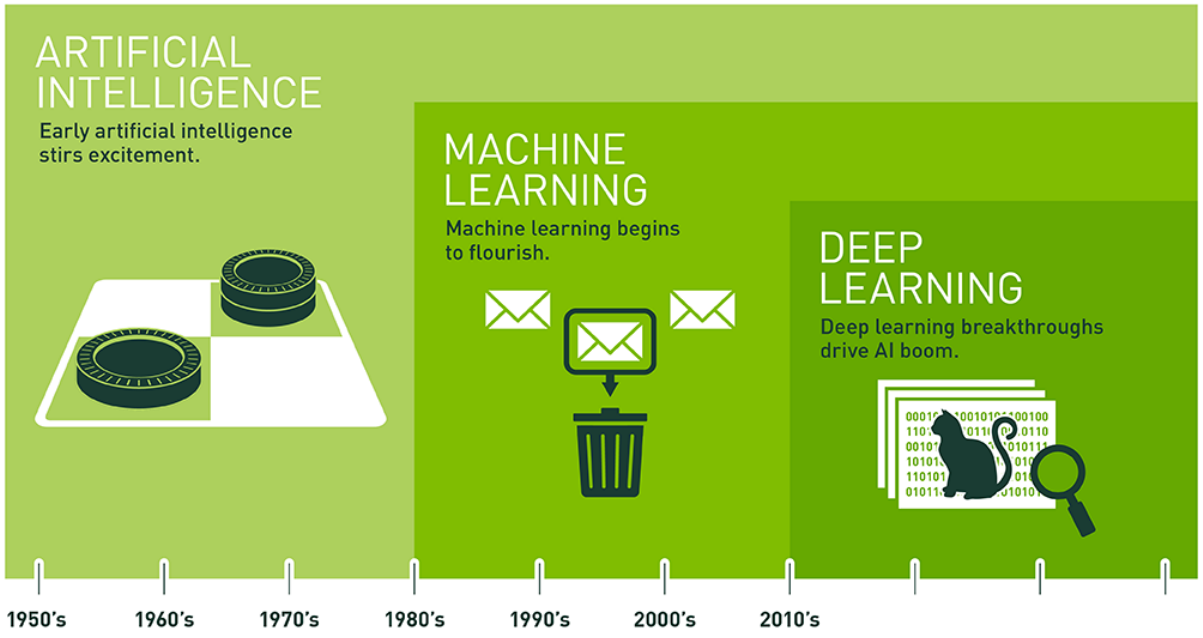
Where we are now..



Where we are now..

Deep Learning

- Deep learning refers to artificial neural networks that are composed of many layers.
- It’s a growing trend in Machine Learning due to some favorable results in applications where the target function is very complex and the datasets are large.



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Deep Learning: Strengths

- **Robust**
  - No need to design the features ahead of time - features are automatically learned to be optimal for the task at hand
  - Robustness to natural variations in the data is automatically learned
- **Generalizable**
  - The same neural net approach can be used for many different applications and data types
- **Scalable**
  - Performance improves with more data, method is massively parallelizable

Deep Learning: Weaknesses

- Deep Learning **requires a large dataset**, hence long training period.
- In term of cost, Machine Learning methods like SVMs and other tree ensembles are very easily deployed even by relative machine learning novices and can usually get you reasonably good results.
- Deep learning methods **tend to learn everything**. It's better to encode prior knowledge about structure of images (or audio or text).
- The learned features are often **difficult to understand**. Many vision features are also not really human-understandable (e.g, concatenations/combinations of different features).
- Requires **a good understanding of how to model** multiple modalities with traditional tools.

Deep Learning: Applications

Deep learning libraries

Microsoft CNTK

torch

theano

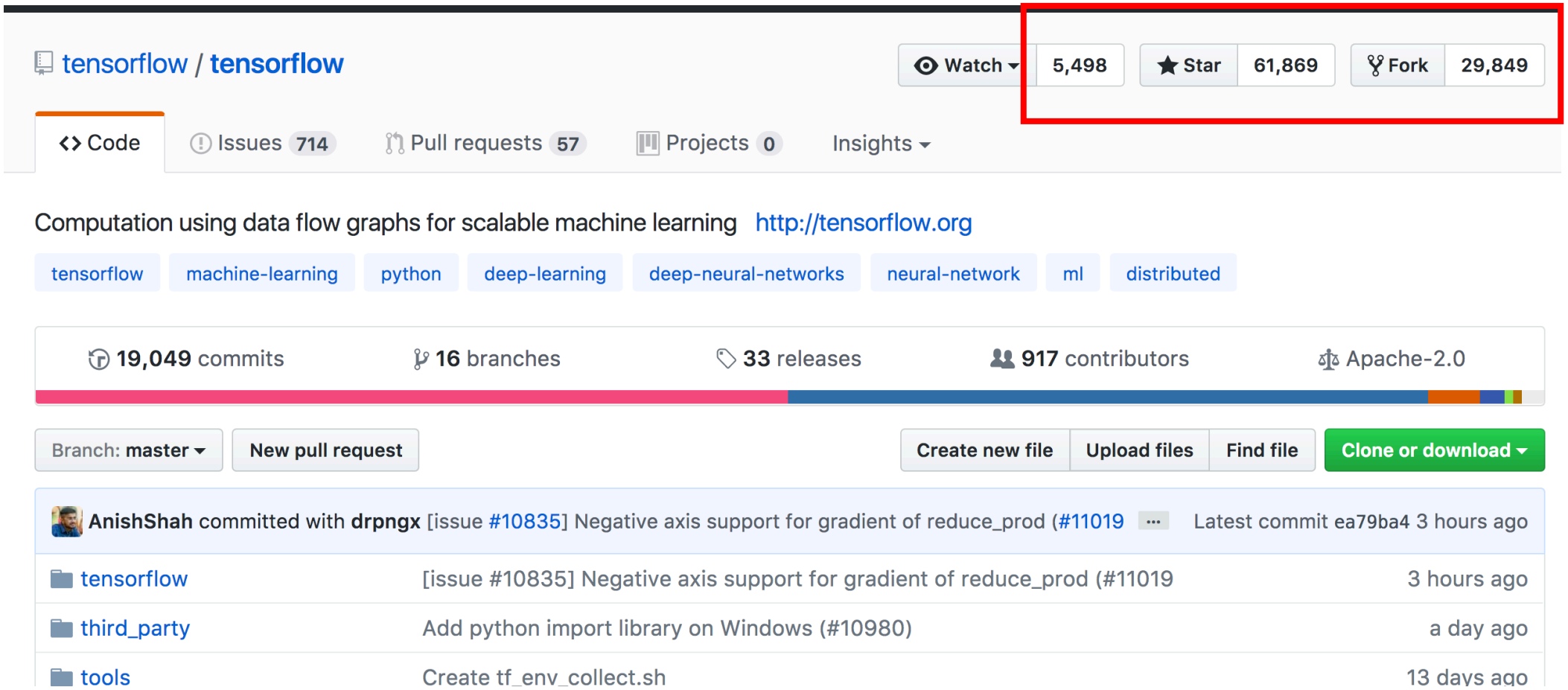
Caffe

mxnet

What is TensorFlow?

- URL: https://www.tensorflow.org/
- Released under the open source license on November 9, 2015
- Current version 1.2
- Open source software library for numerical computation using data flow graphs
- Originally developed by Google Brain Team to conduct machine learning and deep neural networks research
- General enough to be applicable in a wide variety of other domains as well
- TensorFlow provides an extensive suite of functions and classes that allow users to build various models from scratch.

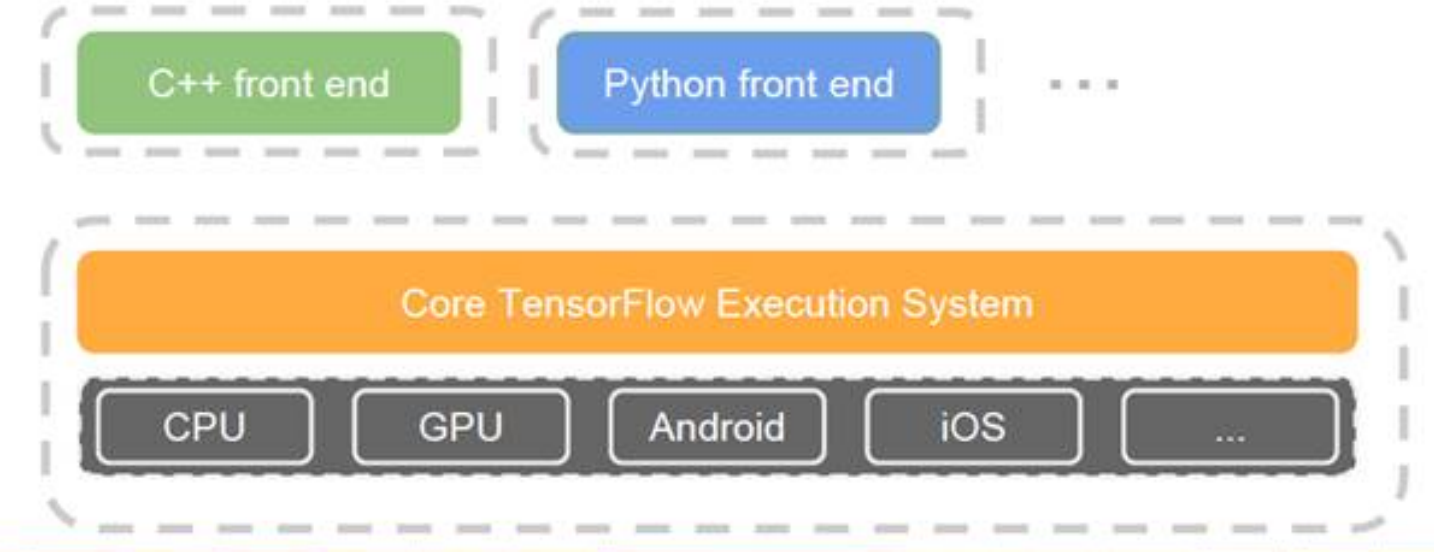
Most popular on Github



https://github.com/tensorflow/tensorflow

TensorFlow architecture

- Core in C++
  - Low overhead
- Different front ends for specifying/driving the computation
  - Python, C++, R and many more



TensorFlow Models

https://github.com/tensorflow/models

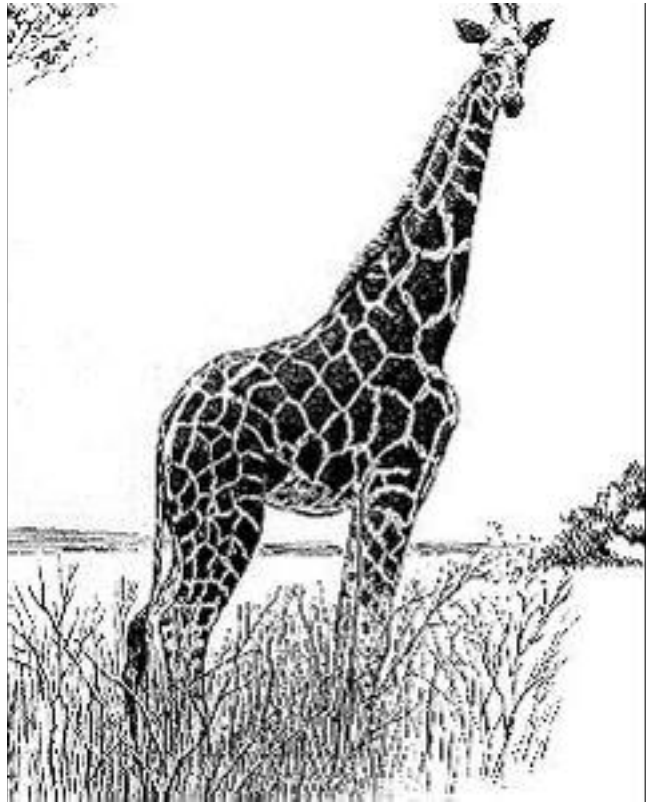
Models

- adversarial_crypto: protecting communications with adversarial neural cryptography.
- adversarial_text: semi-supervised sequence learning with adversarial training.
- attention_ocr: a model for real-world image text extraction.
- autoencoder: various autoencoders.
- cognitive_mapping_and_planning: implementation of a spatial memory based mapping and planning architecture for visual navigation.
- compression: compressing and decompressing images using a pre-trained Residual GRU network.
- differential_privacy: privacy-preserving student models from multiple teachers.
- domain_adaptation: domain separation networks.
- im2txt: image-to-text neural network for image captioning.
- inception: deep convolutional networks for computer vision.

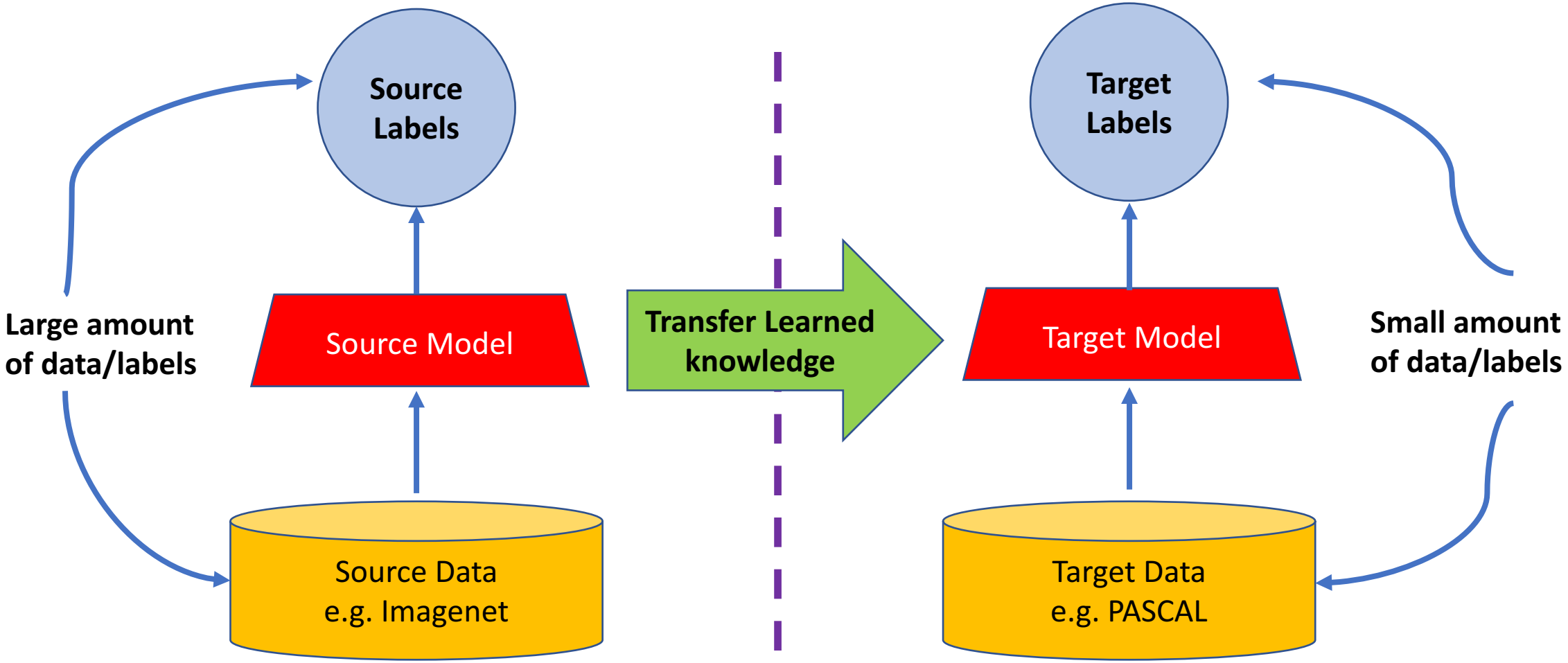
Transfer Learning : Idea

- Instead of training a deep network from scratch for your task:
  - Take a network trained on a different domain for a different **source task**
  - Adapt it for your domain and your **target task**

Image classification: Leopard or Giraffe?

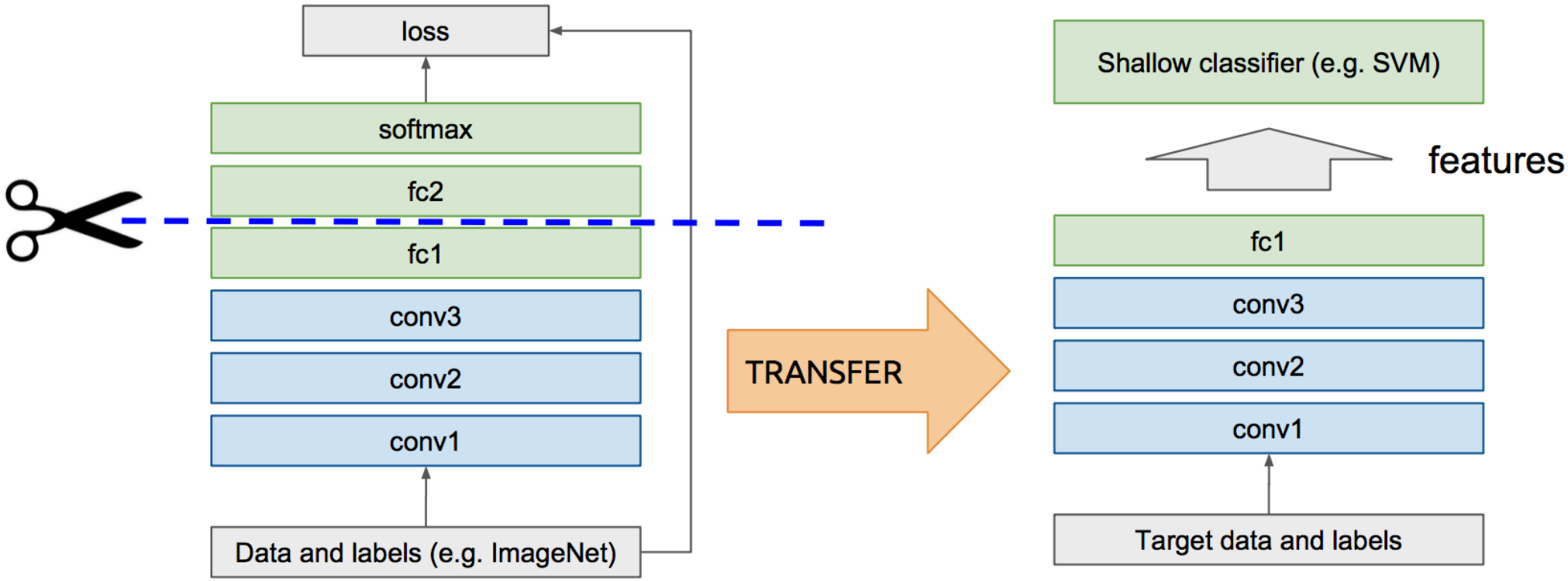


Transfer Learning: Idea



Off the Shelf

- Idea: use outputs of one or more layers of a network trained on a different task as generic feature detectors. Train a new shallow model on these features.



Why use a pre-trained model?

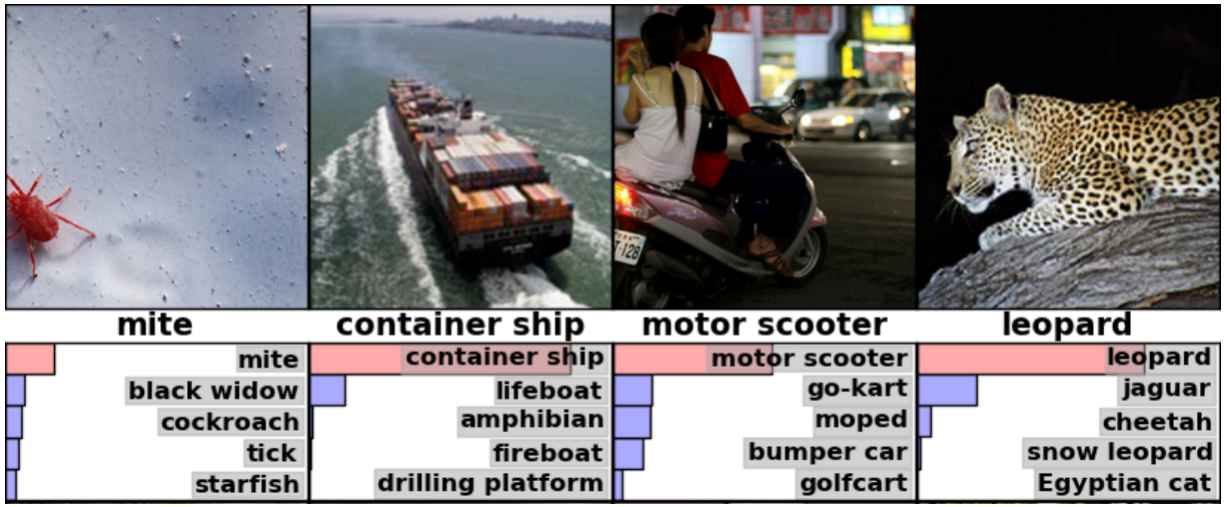
- It is faster (it’s pre-trained)
- It is cheaper (no need to have GPU farm)
- Achieve good accuracy

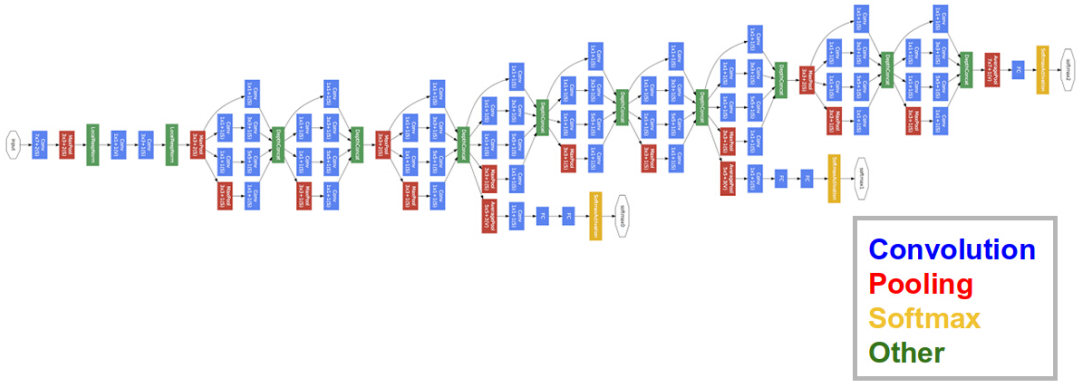
ImageNet

- ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images.
- Currently there are over five hundred images on average per node.

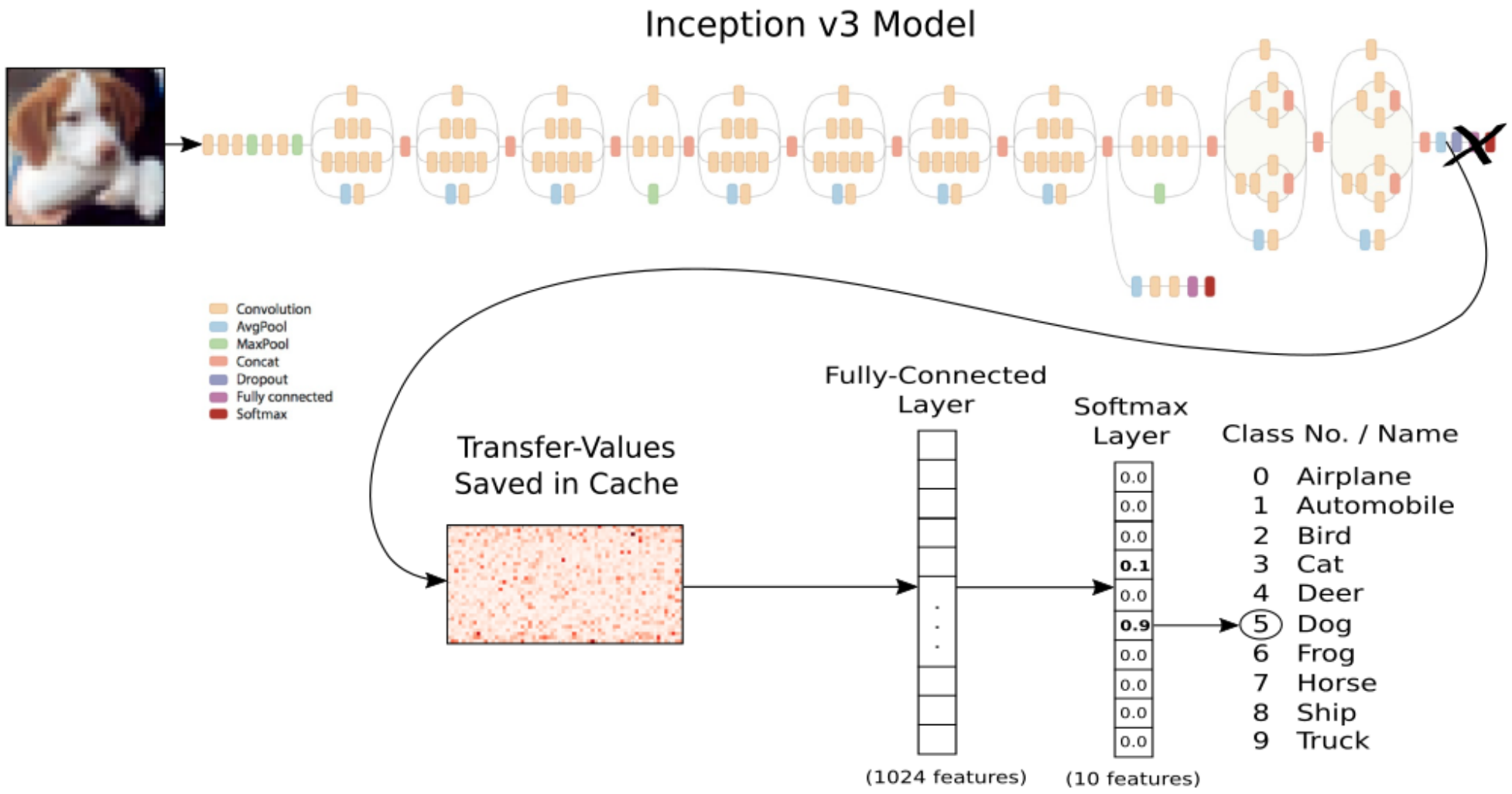
Inception V3

- Inception-v3 is trained for the ImageNet Large Visual Recognition Challenge using the data from 2012.
- This is a standard task in computer vision, where models try to classify entire images into 1000 classes, like "Zebra", "Dalmatian", and "Dishwasher".





Transfer Learning

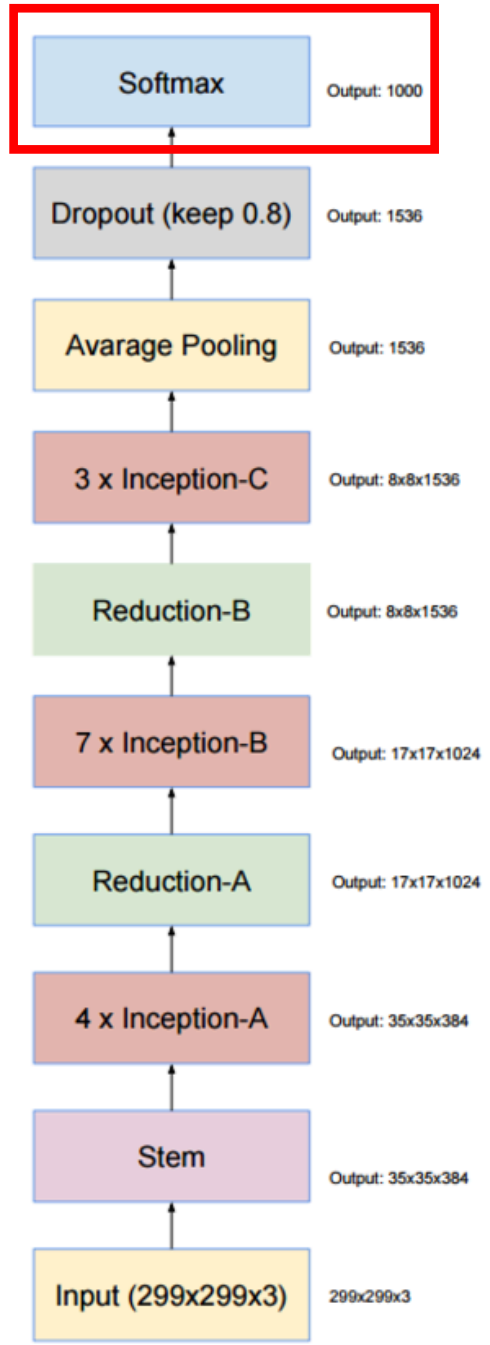


Transfer Learning

- Pre-trained model has learned to pick out features from images that are useful in distinguishing one image (class) from another.
- Initial layer filters encode edges and color, while later layer filters encode texture and shape.
- Cheaper to “transfer” that learning to new classification scenario than retrain a classifier from scratch.

Transfer Learning

- Remove the Fully Connected (Bottleneck layer) from pre-trained ImageNet Inception v3 model.
- Run images from Dataset through this truncated network to produce (semantic) image vectors.
- Use these vectors to train another classifier to predict the labels in training set.
- Prediction
  - Image needs to be preprocessed into image vector through truncated pre-trained ImageNet Inception v3 model.
  - Prediction made with second classifier against image vector



Further Fine Tuning...

- Remove bottleneck (classifier) layer from pre-trained network.
- Freeze all weights except the last (few) convolutional layers.
- Attach your own classifier to the bottom.
- Train the resulting classifier with very low learning rate.
- Computationally more expensive than Transfer Learning but still cheaper than training network from scratch.
- More robust model.

Transfer Learning with TensorFlow

- Transfer learning does not require GPUs to train
- Training across the training set (2,000 images) took less than a minute on my Macbook Pro without GPU support. This is not entirely surprising though, as the final model is just a softmax regression.
- With TensorBoard, it is able to provide summaries that make it easier to understand, debug, and optimize the retraining.
- It is also able visualize the graph and statistics, such as how the weights or accuracy varied during training.
- https://www.tensorflow.org/tutorials/image_retraining

Demo

Slides & Codes available from Github:

http://bit.ly/google-io-kl

Summary

- Possible to train very large models on small data by using transfer learning and domain adaptation
- Off the shelf features work very well in various domains and tasks
- Lower layers of network contain very generic features, higher layers more task specific features
- Supervised domain adaptation via fine tuning almost always improves performance

Thanks!

Questions?

@kuanhoong

https://www.linkedin.com/in/kuanhoong

kuanhoong@gmail.com