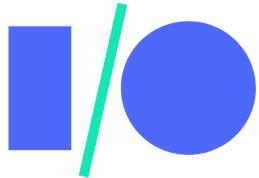


Explore and have fun with TensorFlow



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



About me



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



- Senior Data Scientist



- Senior Manager Data Science



- 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.



TensorFlow & Deep Learning Malaysia

Public group

Discussion

Members

Events

Photos

Files

Manage Group

Search this group

PINNED POST



Kuan Hoong created a poll.



Joined

Notifications

Share

...

Write Post

Add Photo/Video

Live Video

More



Write something...



Photo/Video



Feeling/Activity

...

Shortcuts

TensorFlow & Deep Lea...

R User Group Malaysia 4

R User Group Malaysia ...

TensorFlow KR 11

TensorFlow India

Github: <https://github.com/TensorFlowMY>

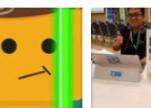
ADD MEMBERS

+ Enter name or email address...



MEMBERS

256 members (77 new)



SUGGESTED MEMBERS



Kurt Orion G

Add Member

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



TensorFlow and Deep Learning Malaysia

[Home](#)[Members](#)[Sponsors](#)[Photos](#)[Pages](#)[Discussions](#)[More](#)[Group tools](#)[My profile](#)[Change photo](#)

Kuala Lumpur,
Malaysia

Founded Jul 4, 2017

[About us...](#)[+ Invite friends](#)

Welcome!

[+ Schedule a new Meetup](#)

Upcoming (1) [Calendar](#)

TensorFlow & Deep Learning Malaysia Inaugural Meetup

ASEAN Data Analytics Exchange (ADAX),
Level 27, Tower B, The Vertical Business Suite,
Avenue 3, Bangsar South, 59200, Kuala Lumpur ([map](#))



The TensorFlow and Deep Learning Malaysia welcomes you to the inaugural meet-up for the month of July 2017. The group's aim is to enable people to create and deploy...

[Learn more](#)

Thu Jul 6

7:00 PM

I'm going

2 going

0 comments

What's new

NEW MEMBER

[Yap Wen Jiun](#)
joined



Yesterday

NEW MEMBER

[Ng Chin Kit](#) joined



Yesterday

NEW MEMBER

[Vishnu Monn](#)
joined



Yesterday

NEW RSVP

[John See](#) RSVPed
Yes for [TensorFlow & Deep Learning Malaysia Inaugural Meetup](#)



Yesterday

<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

[Schedule a Meetup](#)

You can schedule one!

Recent Meetups

Oct 20, 2016 · 7:00 PM

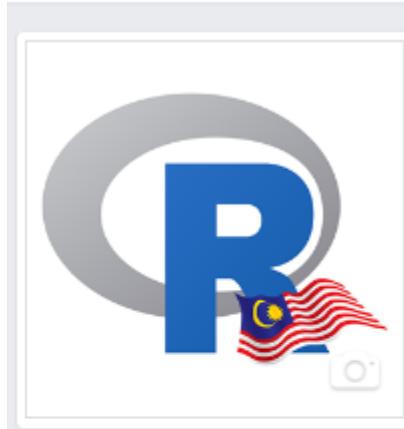
[Rate this Meetup](#)

What's new



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





R User Group Malaysia

@rusergroupmalaysia

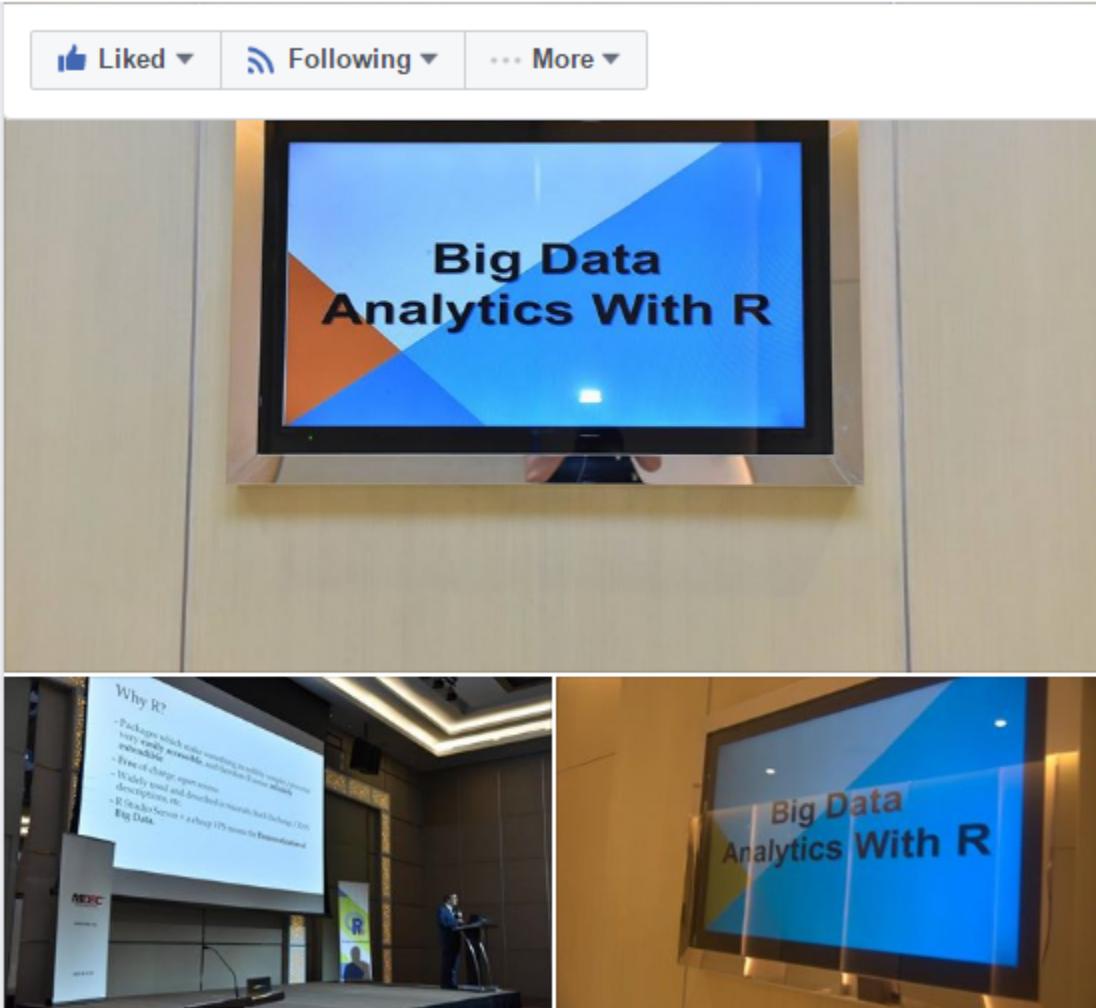
Home

About

Photos

Events

Likes



Like Liked ▾

Following ▾

More ▾

Learn More



224 likes 0 this week

Andy Low and 18 other friends

226 follows

See Pages Feed

Posts from Pages you've liked as your Page

Invite friends to like this Page

309 post reach this week

The R User Group Malaysia is a diverse group that come together to discuss anything related to the R programming language.



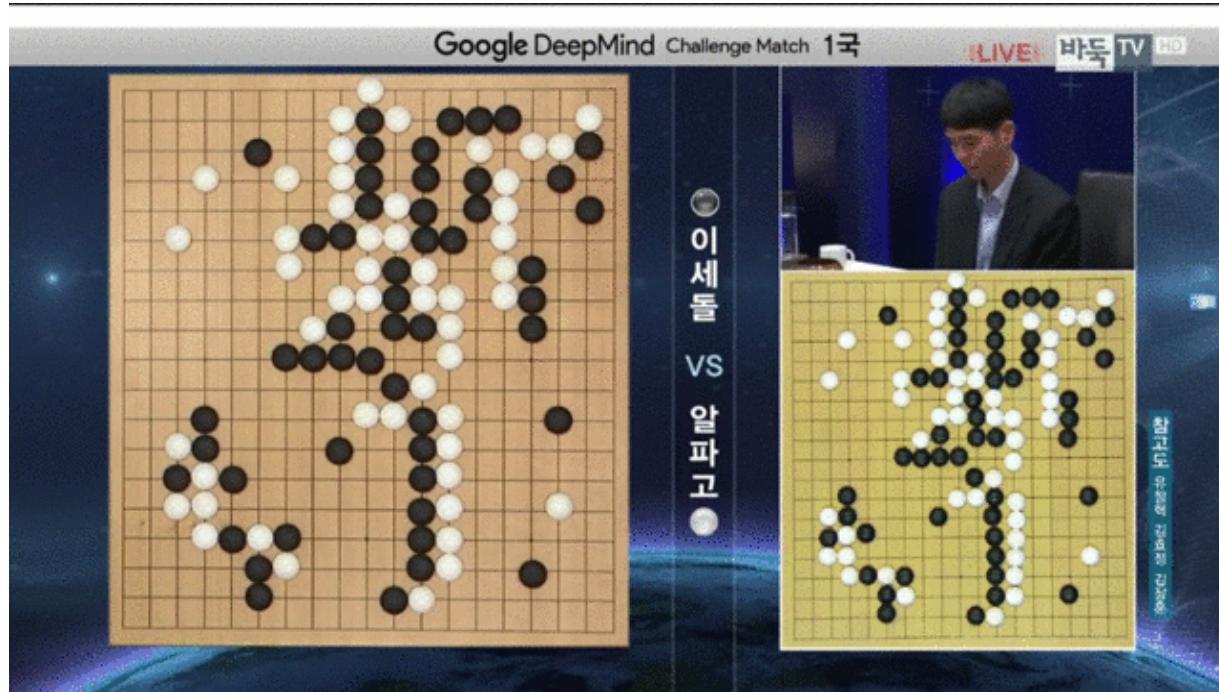
224 Likes

Andy Low and 18 other friends like this

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



Where we are now..

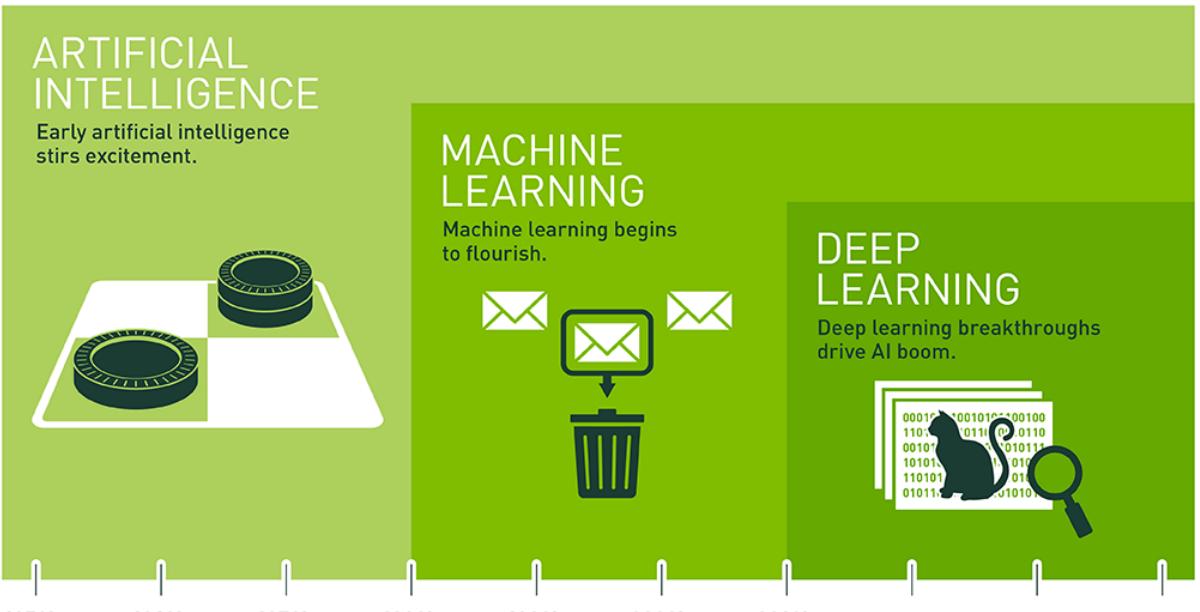


Where we are now..



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

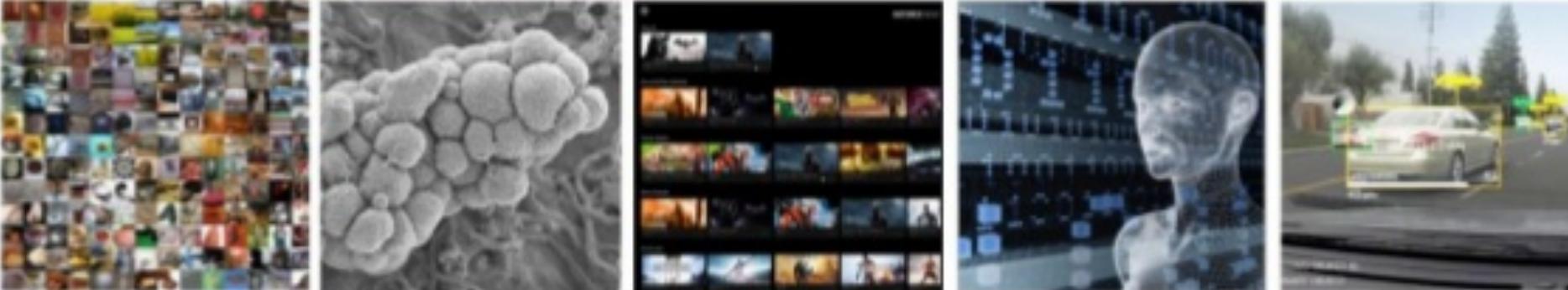
- **Robust**
 - No need to design the features ahead of time - features are automatically learned to be optimal for the task at hand
 - Robustness to natural variations in the data is automatically learned
- **Generalizable**
 - The same neural net approach can be used for many different applications 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 EVERYWHERE



INTERNET & CLOUD	MEDICINE & BIOLOGY	MEDIA & ENTERTAINMENT	SECURITY & DEFENSE	AUTONOMOUS MACHINES
Image Classification Speech Recognition Language Translation Language Processing Sentiment Analysis Recommendation	Cancer Cell Detection Diabetic Grading Drug Discovery	Video Captioning Video Search Real Time Translation	Face Detection Video Surveillance Satellite Imagery	Pedestrian Detection Lane Tracking Recognize Traffic Sign

Deep learning libraries



theano



Caffe

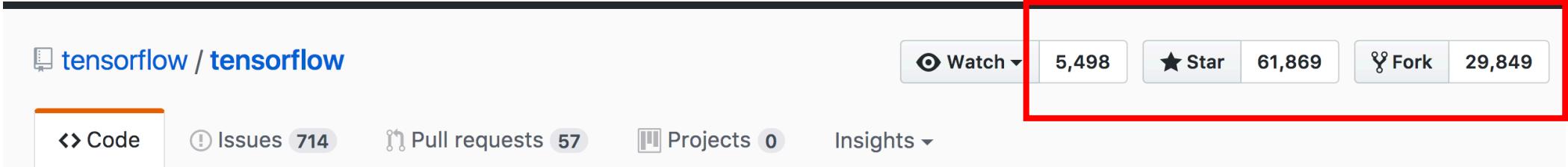


What is TensorFlow?

- 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



tensorflow / tensorflow

Watch 5,498 Star 61,869 Fork 29,849

Code Issues 714 Pull requests 57 Projects 0 Insights

Computation using data flow graphs for scalable machine learning <http://tensorflow.org>

tensorflow machine-learning python deep-learning deep-neural-networks neural-network ml distributed

19,049 commits 16 branches 33 releases 917 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

AnishShah committed with drpngx [issue #10835] Negative axis support for gradient of reduce_prod (#11019) ... Latest commit ea79ba4 3 hours ago

tensorflow [issue #10835] Negative axis support for gradient of reduce_prod (#11019) 3 hours ago

third_party Add python import library on Windows (#10980) a day ago

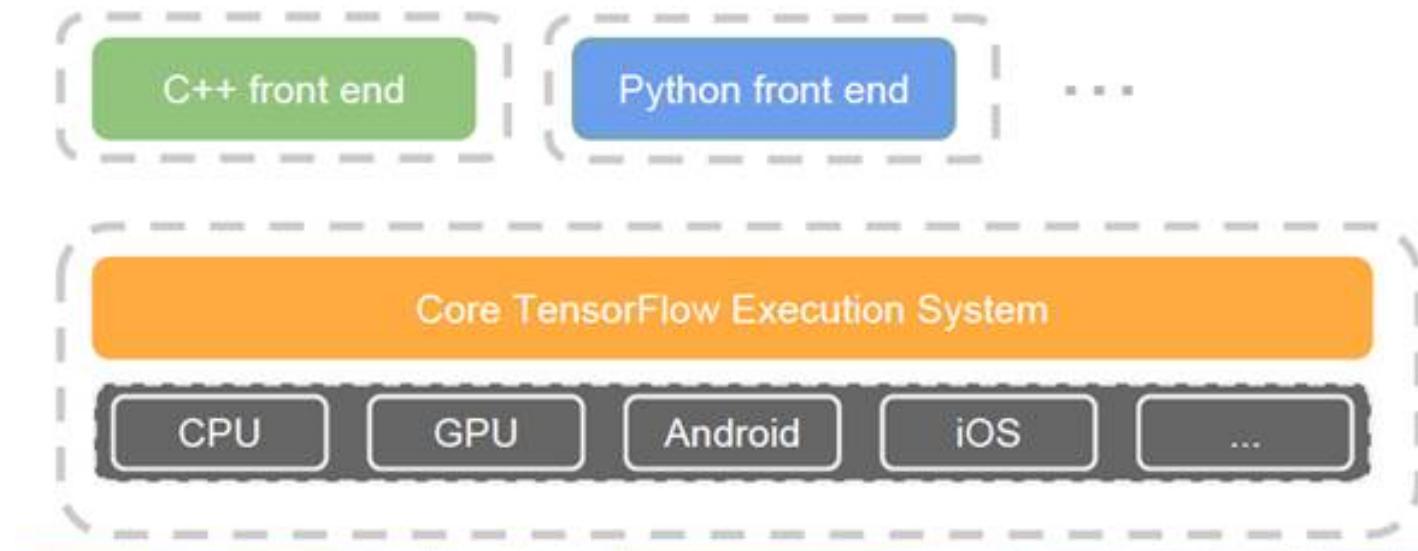
tools Create tf_env_collect.sh 13 days ago

<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>

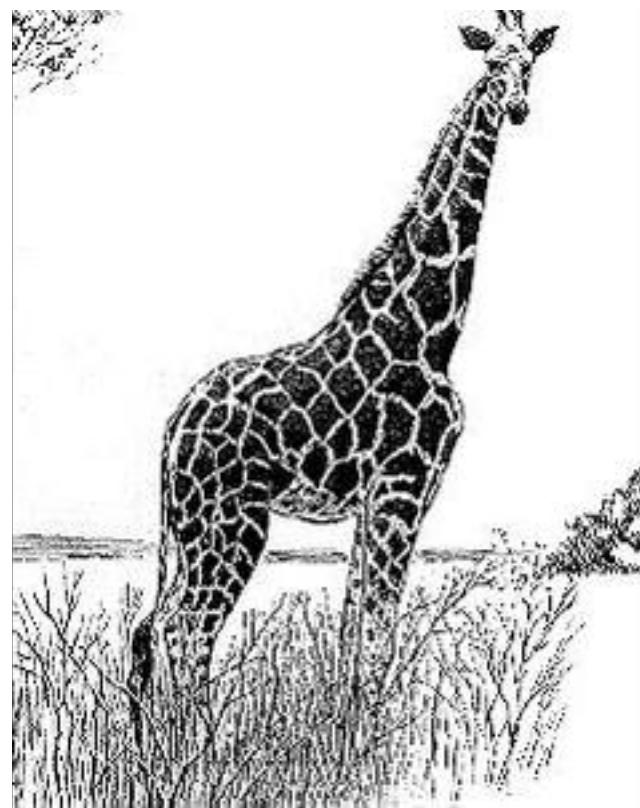
Models

- [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.

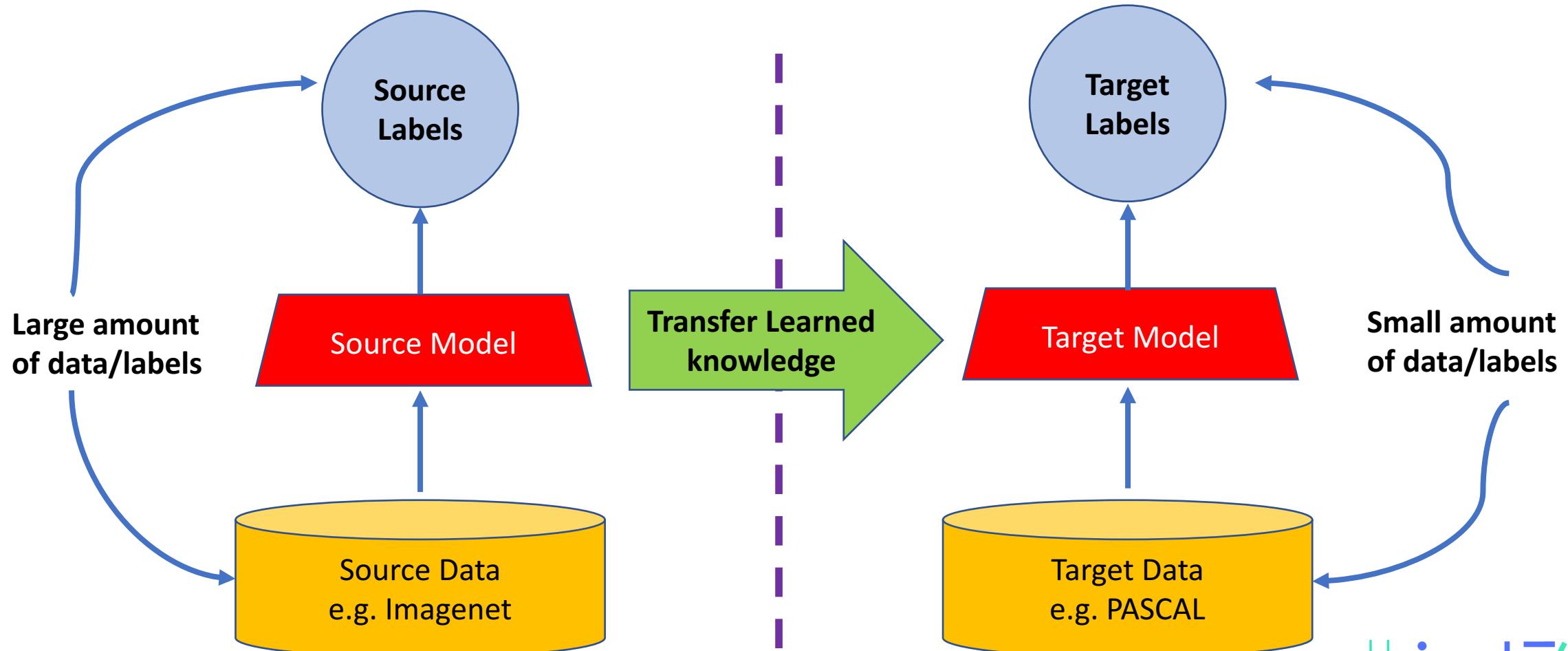
Transfer Learning : Idea

- 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?

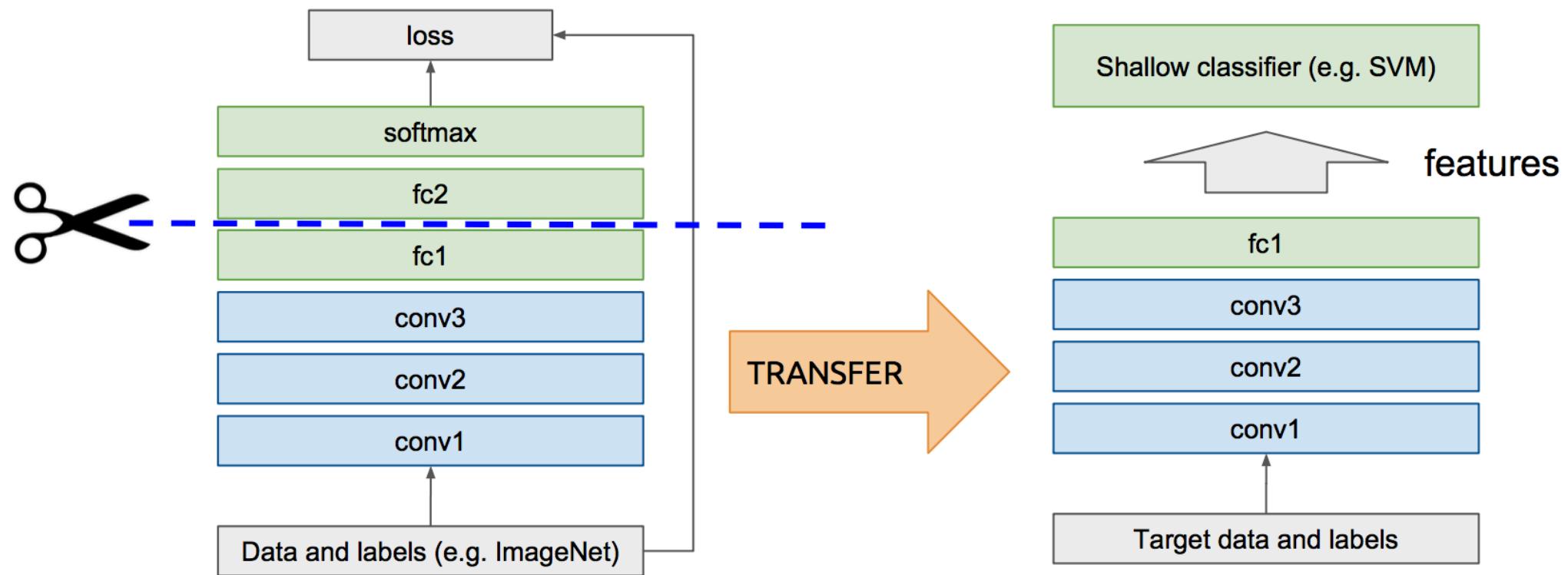


Transfer Learning: Idea



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.

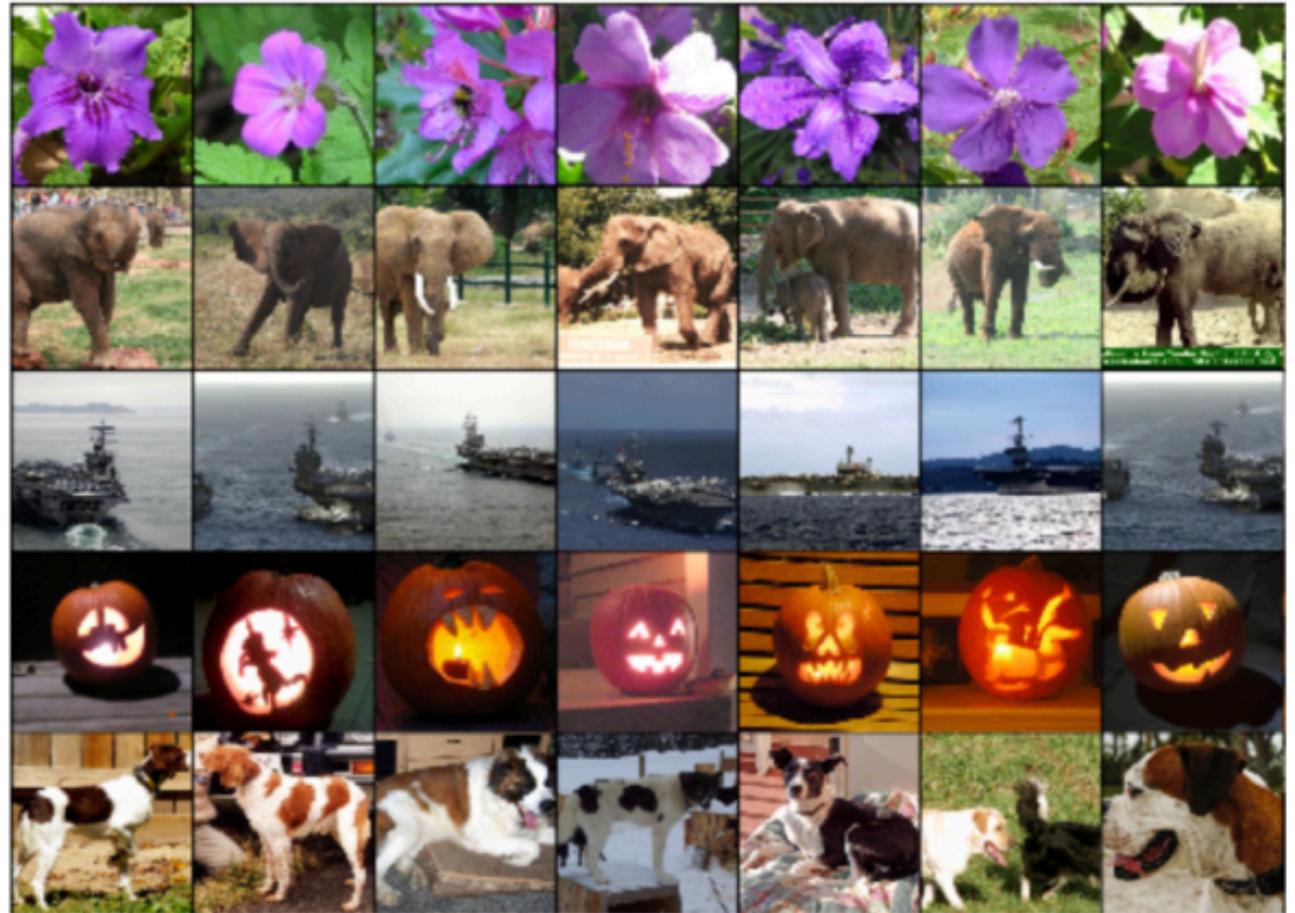


Why use a pre-trained model?

- 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.

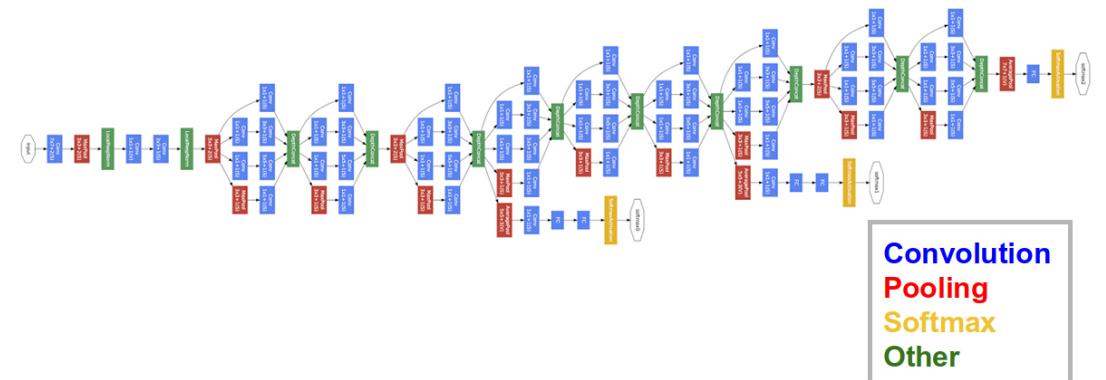


Inception V3

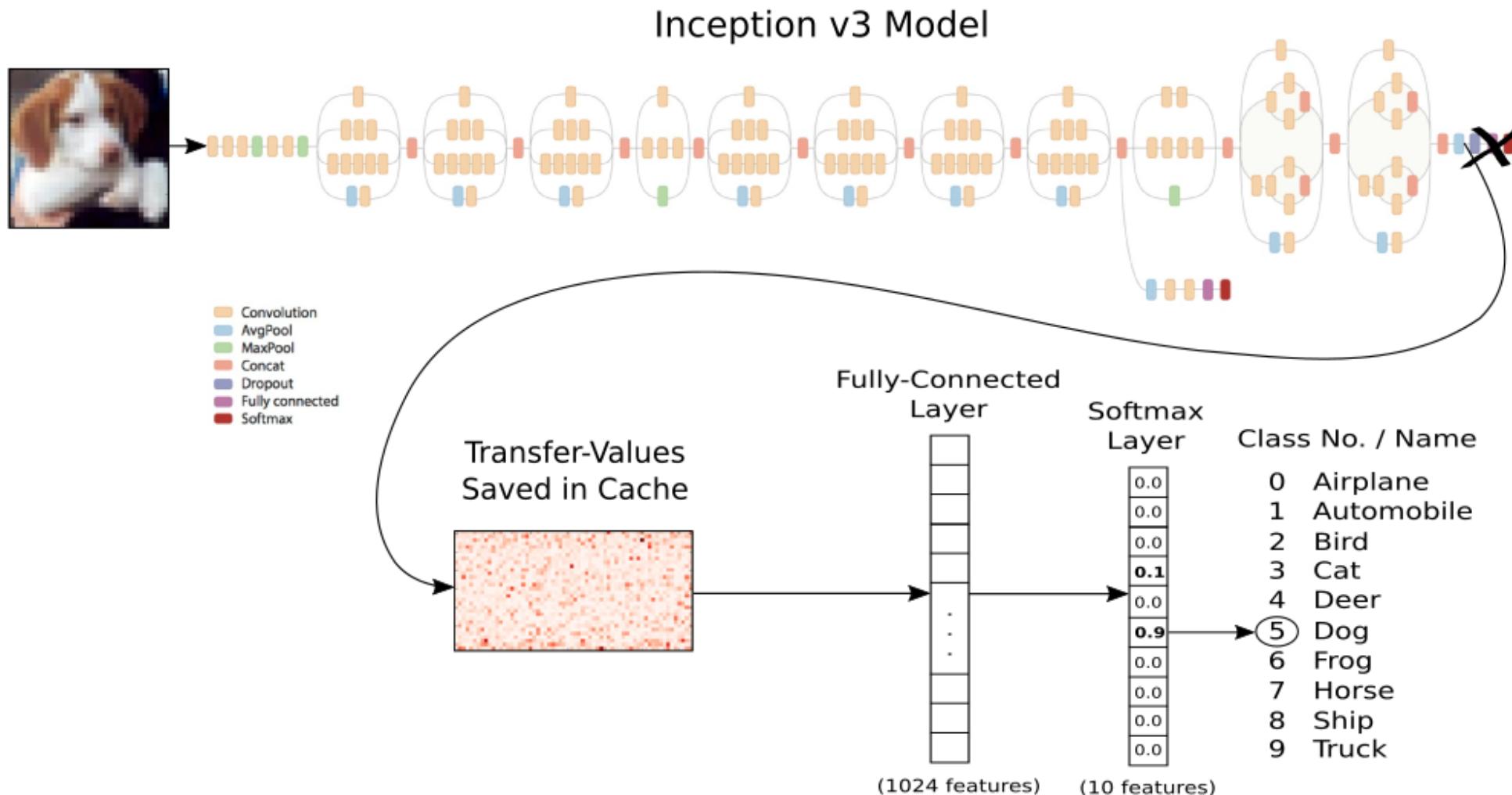
- [Inception-v3](#) is trained for the [ImageNet](#) Large Visual Recognition Challenge using the data from 2012.
- This is a standard task in computer vision, where models try to classify entire images into [1000 classes](#), like "Zebra", "Dalmatian", and "Dishwasher".



mite	container ship	motor scooter	leopard
mite	container ship	motor scooter	leopard
black widow	lifeboat	go-kart	jaguar
cockroach	amphibian	moped	cheetah
tick	fireboat	bumper car	snow leopard
starfish	drilling platform	golfcart	Egyptian cat



Transfer Learning

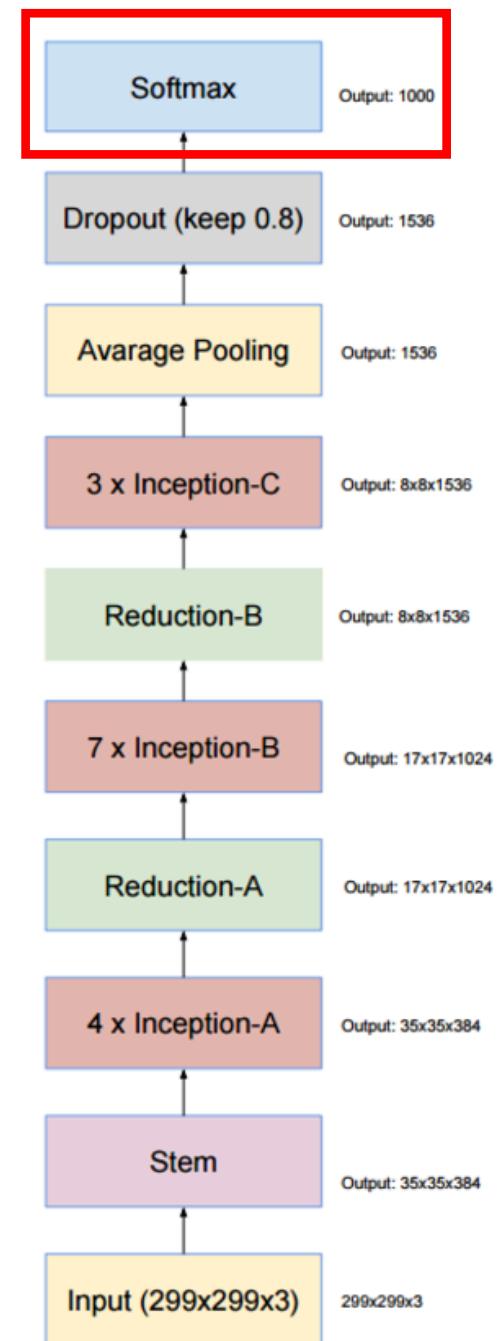


Transfer Learning

- 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?



@kuanhooong



<https://www.linkedin.com/in/kuanhooong>



kuanhooong@gmail.com