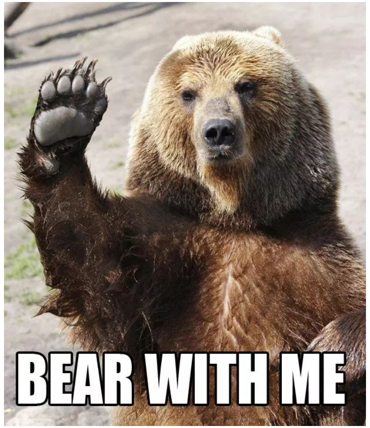
TECHNISCHE UNIVERSITÄT DRESDEN

DRESDEN concept

Learning Compression for High Dimensional Data on the Edge

What?!

This seems far away and not have anything to do with cars?

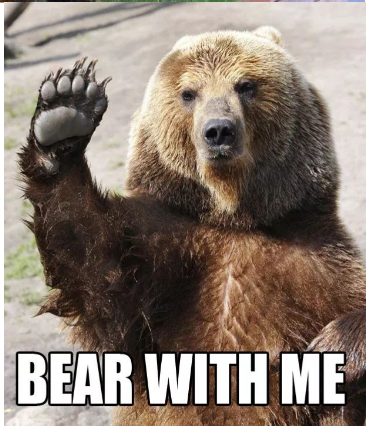


What!

Working Example:

- Cars generate enormous amounts of data
    - 4 TB/hour for average car
- HD Maps needed for navigation to be updated all the time or at least periodically

- Traditional image compression (BPG, i.e. HEVC for stills) is not competitive anymore in terms of compression capacity!
- Image compression will soon also not be competitive with respect to computational (and by extension, electrical) power consumption
    - TPUs or NPUs allow for efficient inference
    - Architectural enhancements: SqueezeNet, NASNet etc.
    - Clever ideas upcoming



Brian Krzanich, CEO Intel - Driven by Data - AutoMobility LA: https://www.youtube.com/watch?v=EskMldJrJdk

Storytime

High dimensional data (images, radar, lidar, etc.) is needed for autonomous, concerted actions

Cars, smart infrastructure & cities, robots, entertainment, ...

Works also for time series: spectrograms

Hypothesis: the fastest bit to transfer is the one we do not need to transfer at all

JPEG is hopelessly outdated and *is being replaced silently*

I will not talk about

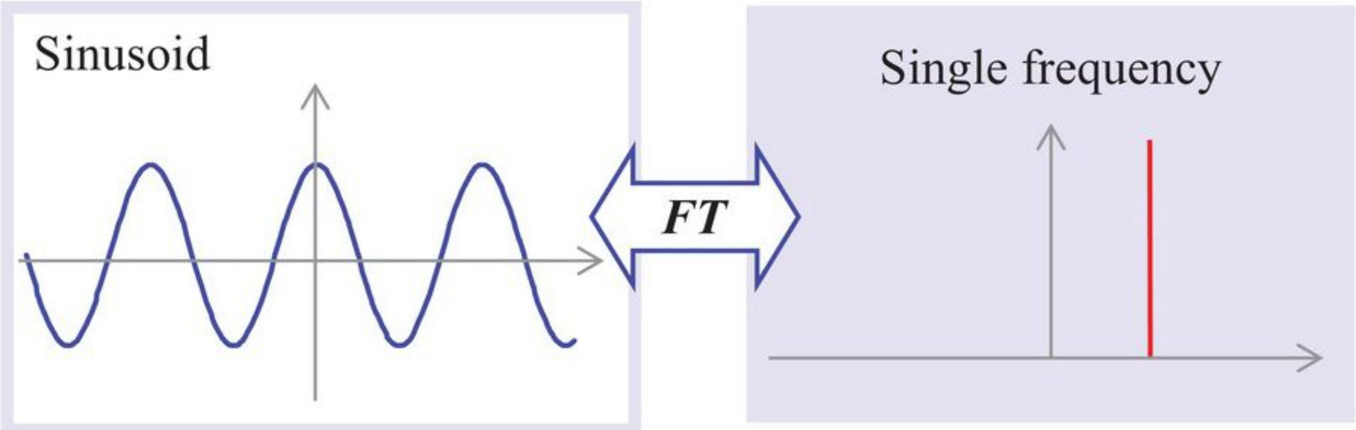
- Bayesian Variational Methods
- BackProp / mini-batching / regularising
- Technical details like Hyperparameter search / optim
- Implementation frameworks (TF, Keras, theano et al.)

The Problem

Representation matters

- Arabic: 123
- Roman: CXXIII
- Binary: 1111011
- Plain: one hundred and twenty three
- Next: 3230

Or of course...



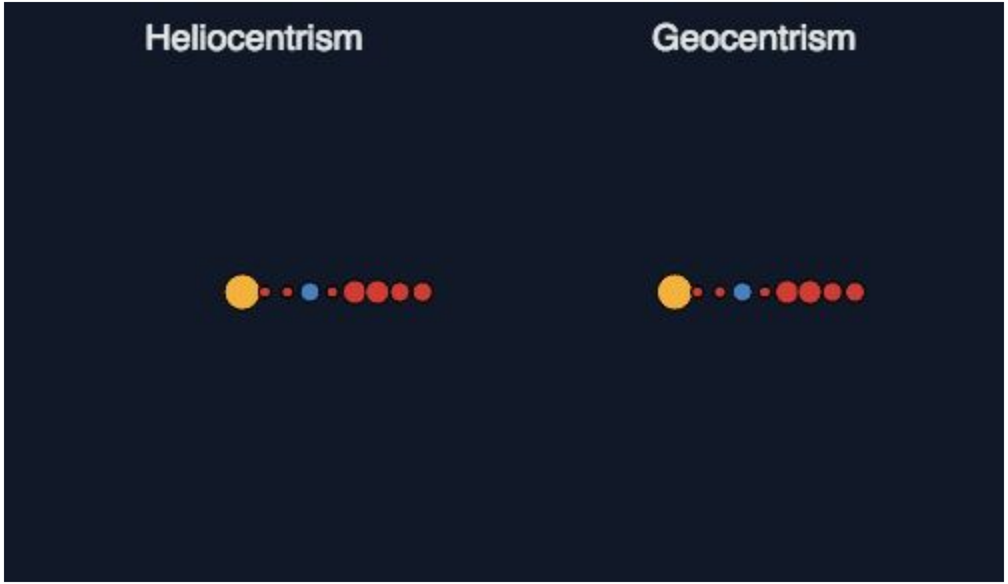
The Problem

Representation matters

- Note the natural model is simpler
- Occam's razor dictates it is more (likely to be) correct

This leads to modelling the world based on observed data

- Perfect model. understand world
- Model expressivity vs. regularisation
- Underlying processes may be much less dimensional (“simpler”)



Machine Learning is {Representation, Feature} Learning

Feature: something that helps draw a (potentially actionable) conclusion

- Previously e.g. Fisher information, Mutual Entropy, etc.; i.e.: hand designed

With the previous slide in mind, the task becomes simply:

- Learn an efficient representation of observed data ("Statistics")
- Either:
  - Hope that what has been observed is representative (hence, "big data")
- Or:
  - Stay agile (hence, "AI". ML is not AI; RL opens opportunities)
- Learning: calibrate / fine tune parameters

  given data

  Transformed problem from a recognition problem to an optimisation problem

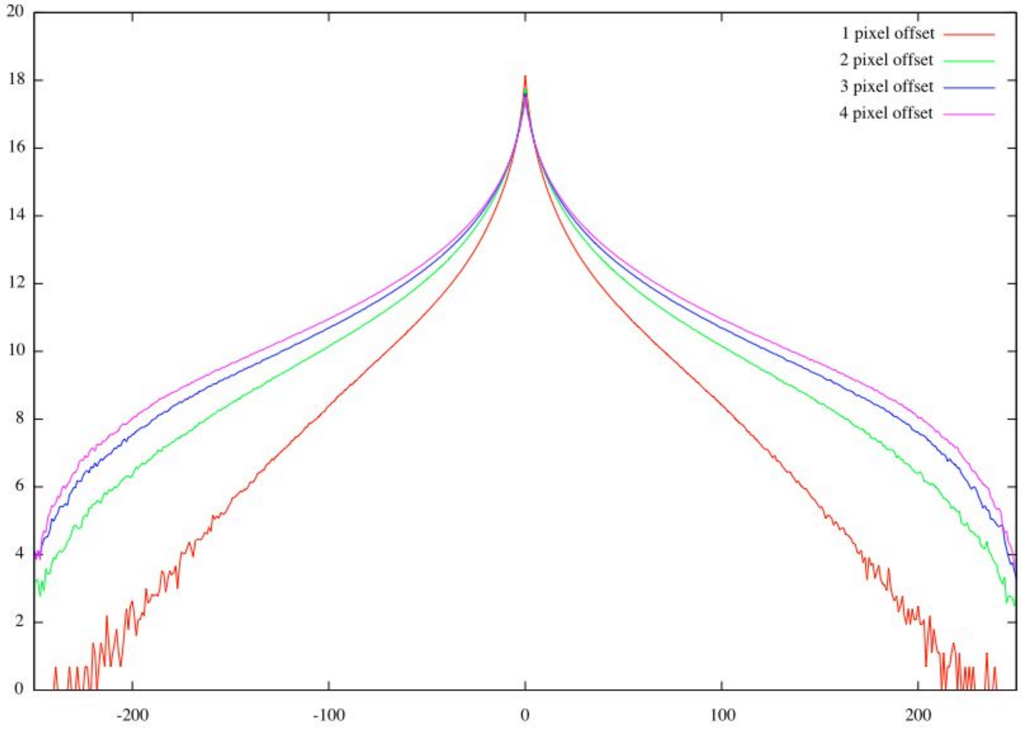
The underlying data is sampled from a probability distribution. The system can either model this distribution ("generative models") or model the desired behaviour given a sample ("conditional models")

Statistics of Natural Images

log-Difference to neighbouring pixels

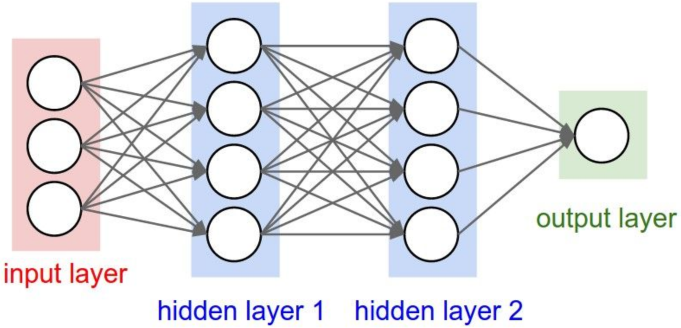
Natural images are smooth

Tail heavy - nothing like a Gaussian



Histogram courtesy of U.Schmidt

Neural Networks



Neural Network* crash course

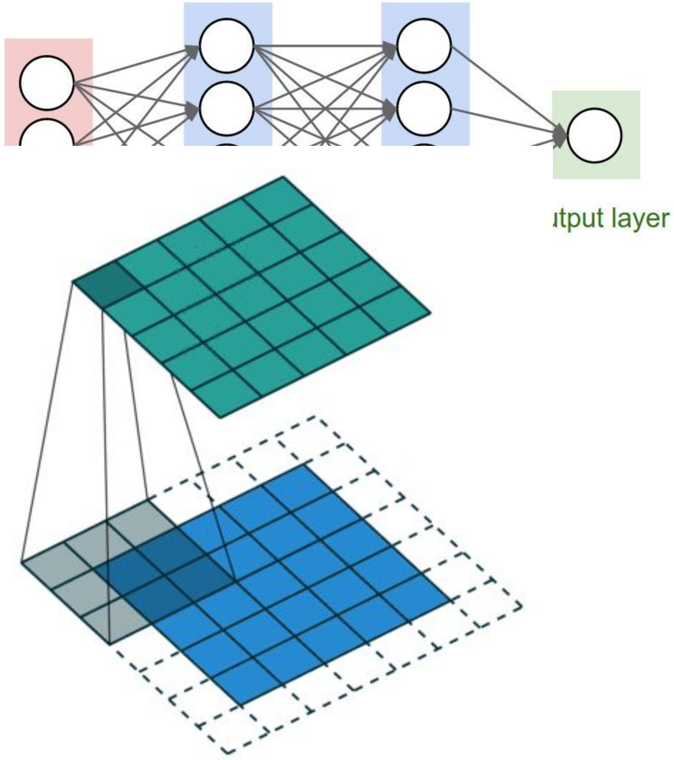
- Iteratively computes nonlinear function of weighted input sums in forward pass
- Learning: optimise weights s.t. Optimal w.r.t. objective
- After training, for arbitrary function f : X -> Y, there exists a NN that can approximate f to arbitrary degree
- Hidden spaces may be much higher dimensional than input space
- Structure may be designed
- Structure may be learned (better results)
- Typically: deeper models lead to better results
  - Overfitting vs. Expressivity

That whole thing is one huge mapping operator and potentially

- Highly nonlinear
- Extremely high dimensional

*: (feed- forward)

Neural Networks as Feature Extractors



https://github.com/vdumoulin/conv_arithmetic

Turns out naively connecting everything in a dense manner is intractable

- With n neurons in l layers, this N = (n*n)^l

Introducing convolutions

- Convolves a kernel with input
  - Compact-supp kernel acts on limited part of the input data
  - Kernels are -you guessed it- learned
- They learn representations of data statistics
- i.e. they perform a compression naturally

One input becomes k responses to the k kernels

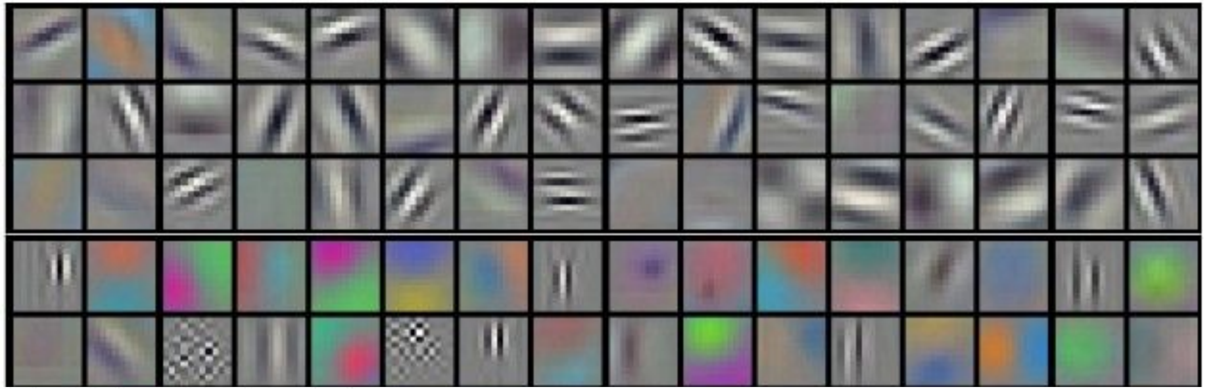
Rinse (pool)

Repeat (conv)

Neural Networks as Feature Extractors

First layer kernel impressions

- Driven by data
- Sensitive to changes
- Naturally finds edges, transitions

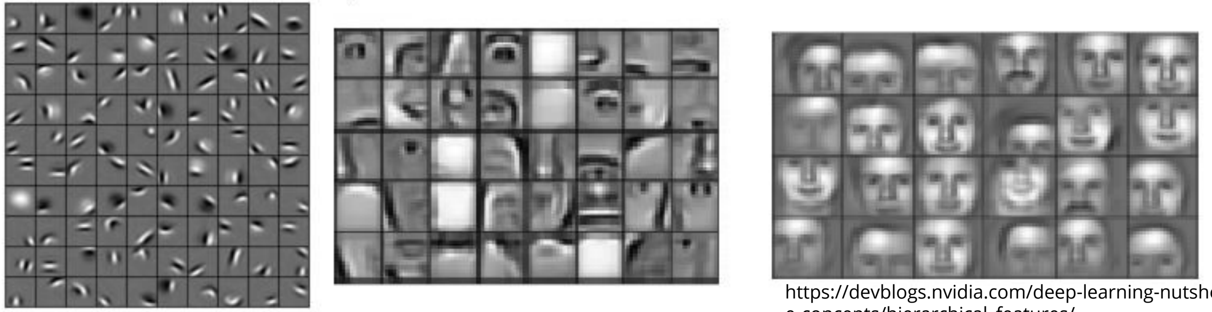


What may the second layer look like?

- Nonlinear, weighted combinations

Of what?

- … everything



https://devblogs.nvidia.com/deep-learning-nutshell-core-concepts/hierarchical_features/

RAISR: Rapid and Accurate Image Super Resolution

Engineered method

1. Cheap and simple upscaling
2. Selecting from a set of learned filters depending on local image content
3. Blend (1) and (2)

   Blending: a learned mapping

Deployed by Google at the end of 2016

Saves ~75 % of image data

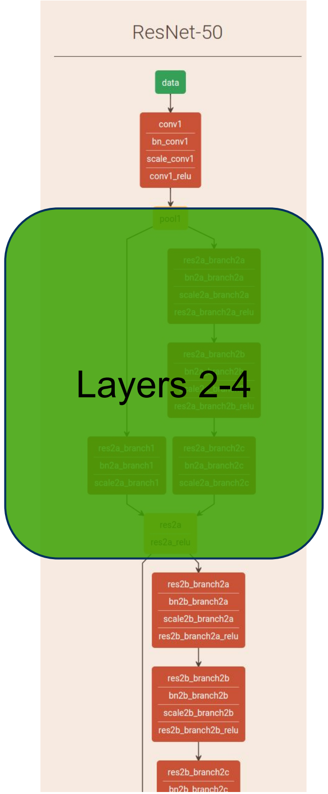
Serves ~ 10^9 images per week

https://ai.googleblog.com/2016/11/enhance-raisr-sharp-images-with-machine.html

Interlude: ResNet-50 Full Network example

ResNet-50 (concurrently with ResNet-152)

Trend to deeper networks: ResNet-1001 exists.

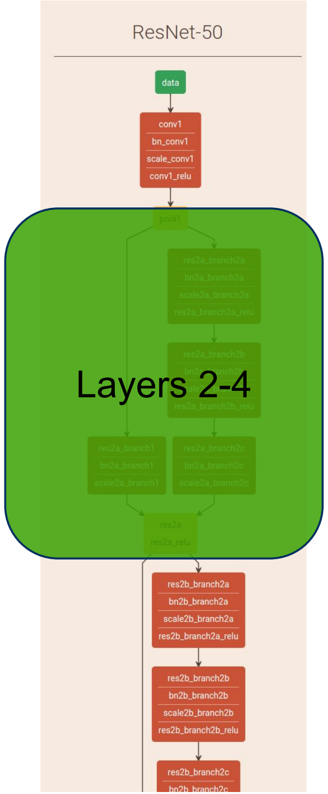


Interlude: ResNet-50 Full Network example

ResNet-50 (concurrently with ResNet-152)

Trend to deeper networks: ResNet-1001 exists.

Google:



OUTRAGEOUSLY LARGE NEURAL NETWORKS: THE SPARSELY-GATED MIXTURE-OF-EXPERTS LAYER

"… [we] present model architectures in which a MoE with up to 137 billion parameters is applied convolutionally… "

… and why? Learning *Compression* on the Edge for …

So we kind of know about feed-forward (convolutional) networks

We have actually looked at a ResNet - so we know about skip connected CNNs

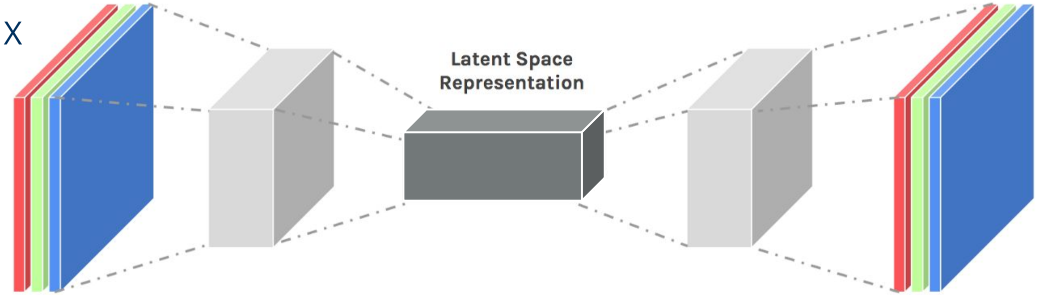
We know they can detect composite features in high dimensional spaces

Visualising or interpreting these may be beyond human understanding

These networks can also classify

- … and localise those classes
- … and can be re-trained to detect new classes with little effort
- .. remember they are actually mappings
- For compression, the relevant mapping is the Identity
    - *Maybe can do nearly as good if only approximate I*
      (Lossy compression)

Autoencoders



A special kind of NN for approximating I: X -> X

Exploit it:

- Input data == Output data
- ... with a bottleneck in between

Force the system to condense data onto a representation that is efficient in a way to allow optimal reconstruction, that is:

compress such that reconstruction is as close to the input as the model parameters allow, without making unwarranted assumptions (such as independence, correlations etc.)

Unwarranted: stay as uniform as possible given the observed data

Model parameters: e.g. a ConvNet for mapping into the latent space, a SConvNet for unwarpping

https://arxiv.org/abs/1801.04260

Autoencoding for lossy image compression

The worst picture

https://arxiv.org/abs/1801.04260

Autoencoding for lossy image compression

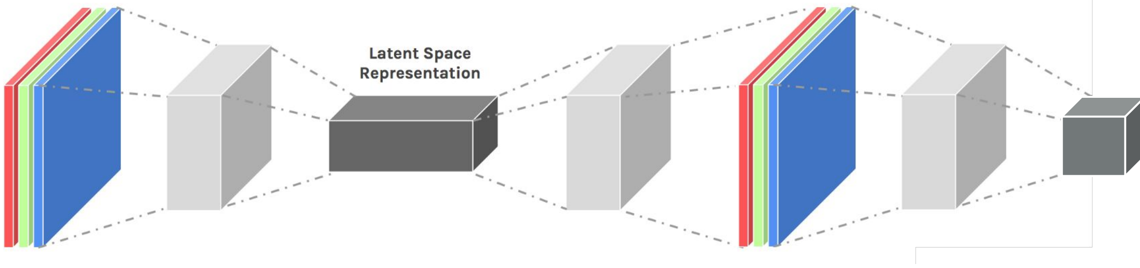
Typical picture

https://arxiv.org/abs/1801.04260

Generative Adversarial Networks

A very special kind of NN for approximating I: X -> X

- Input data == Output data
- … with a bottleneck in between
- Works in a generative manner, where generator is in a competition with the discriminator network



Encoder Generator Discriminator

- Unwrapping of latent representation is done by the generator network
- … which is forced by the discriminator network to improve

State of the Art: Generative Adversarial Networks for Extreme Learned Image Compression

Up to 0.018 bits per pixel

Demo

https://arxiv.org/abs/1804.02958

Further work

Network context

- Break up model:
  - Encoder (on the edge) compresses data Decoder (re-)generates data
- Compare Autoencoders and Generative Models
- Synchronise both parts of the model (for possible online fine tuning)

Demo in context of cars by training on Cityscapes

In terms of cars / intelligent infrastructure: Find the point to branch off from the visual recognition network

- So that compression part is basically free
- This is where the beautiful idea of Inception networks comes into play

https://www.cityscapes-dataset.com

Vision for Cityscapes

Segmentation dataset

ConvNet learns segmentation task, “solves” vision problem

Branching off, small network learns mapping to latent space, encoding spatial & context information

Generator regenerates segmented image

Thanks

Questions?

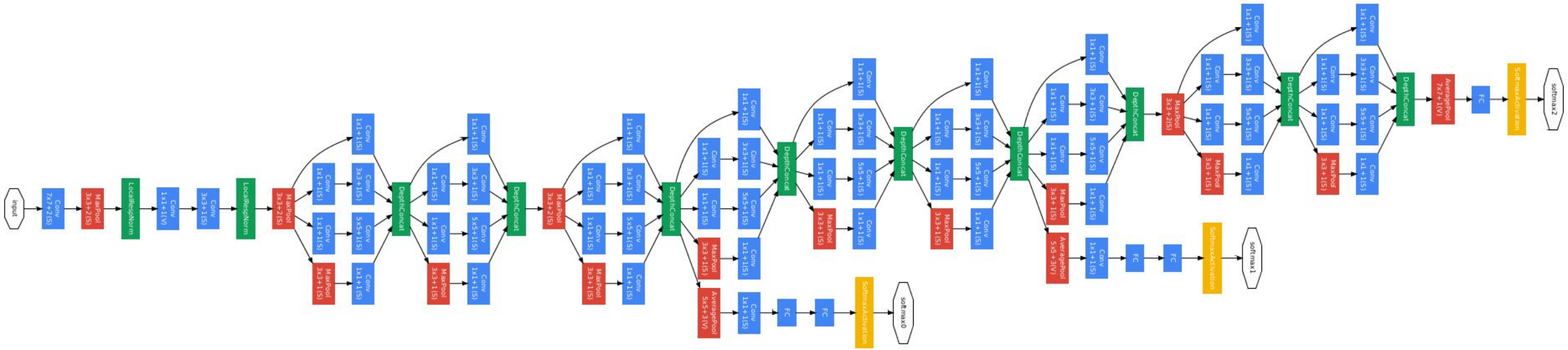
Comments?

Ideas?

Inception / Network in Network

Admittedly, a bit messy

However, best hand designed architecture*



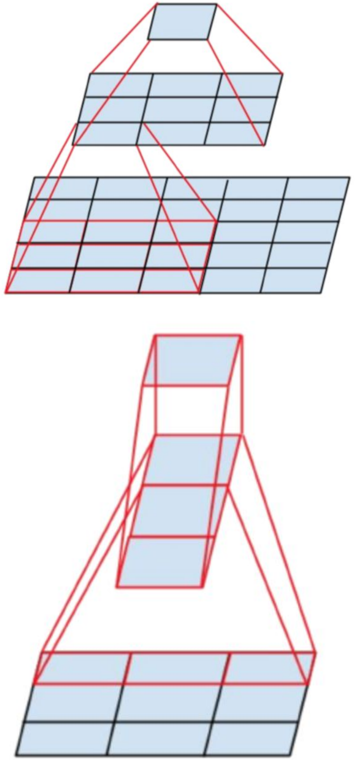
*After including skip connections

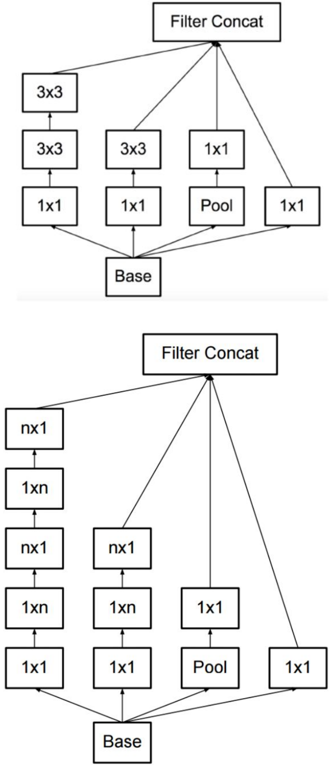
Inception and Xception

Plenty of clever tricks

- Inception introduced composite convolutions
- Xception introduced depthwise separable Convolutions

Both of these save millions of MACCs



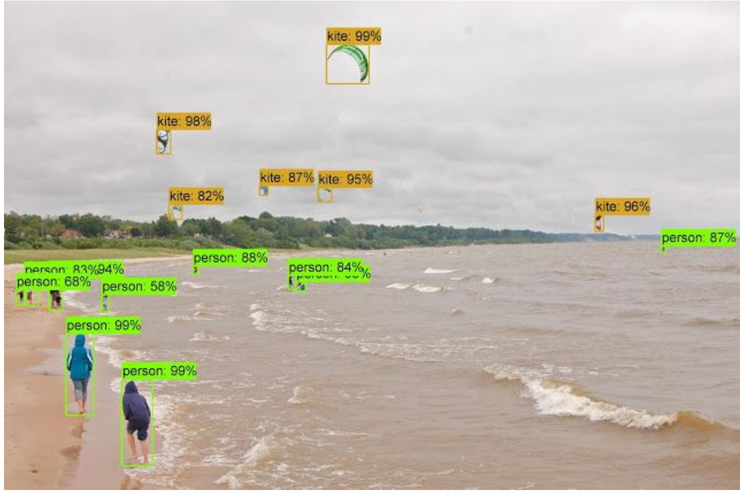


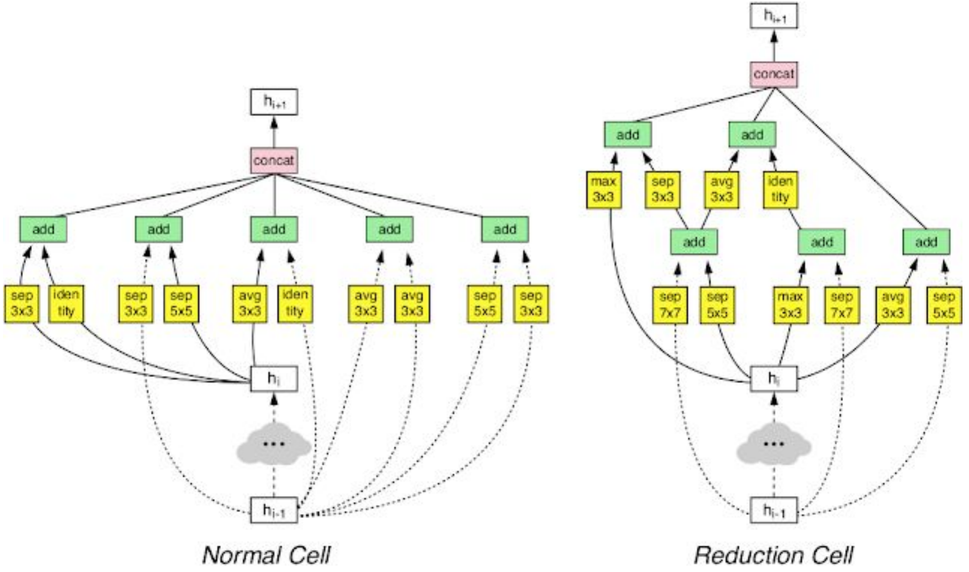
NASNet

Learn not only parameters - learn *design* of network

Introducing AutoML

Beats expert-designed network





https://ai.googleblog.com/2017/11/automl-for-large-scale-image.html

NASNet

"… (our) model is *1.2% better* in top-1 accuracy than the best human-invented architectures while having 9 billion fewer FLOPS - a *reduction of 28% in computational demand* from the previous state-of-the-art model."