Deep Learning Masterclass on Computer Vision

18 March 2017

Who we are

David Low
Chief data scientist at Pand.AI, building AI-powered chatbot to disrupt and shape the booming conversational commerce space with Deep NLP. https://sg.linkedin.com/in/davidlowjw

Jawad
Passionate Data Scientist. Applies advanced analytics and machine learning to Industry use cases. Has keen interest in NLP, recommendation systems and all things data.
https://sg.linkedin.com/in/mdjawad

Wei Min
Data Scientist, Machine Learning expert with experiences in bioinformatics, advanced analytics and data science competitions like Kaggle. https://sg.linkedin.com/in/weimin-wang-ab355325

Zane Lim
Data Scientist, a generalist who dabbles in everything from bayesian statistics to AI.
https://sg.linkedin.com/in/zanelimzy

Agenda

Morning (9am to 12pm)

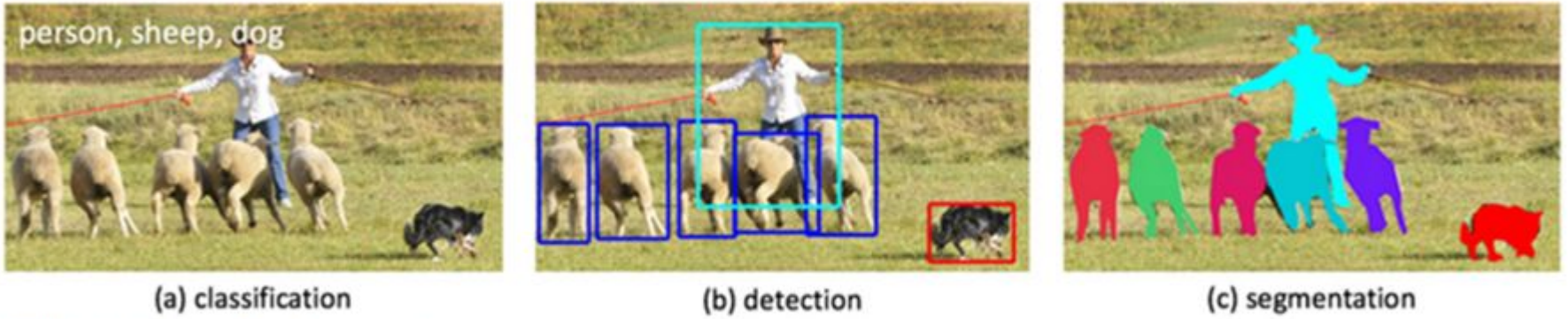
- Computer Vision
  - Traditional approach
  - Deep Learning
- Visualizing the black box
- Convolutional Neural Networks
- Network Architectures
- Transfer Learning
- Deep Learning Frameworks

- Hands-on Workshop
  - Building a CNN based image classifier
- ○ Kaggle in-class challenge starts

Afternoon (1:30pm to 5pm)

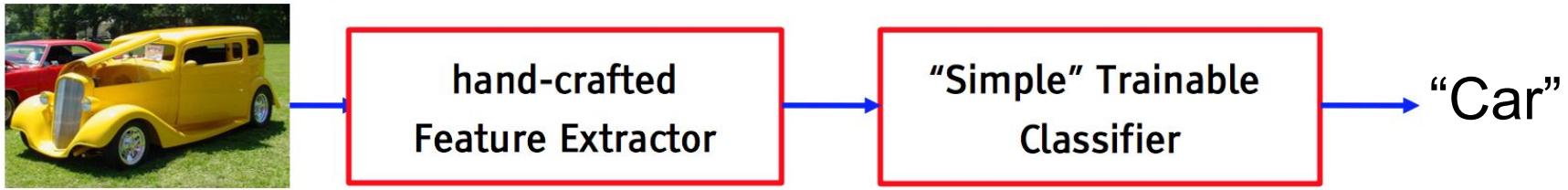
- Real-time Public Leaderboard
- Winner announcement & sharing

Computer Vision



Credit: Tsung-Yi

Traditional Approaches



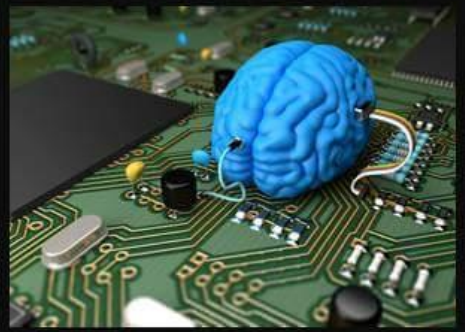
Credit: Ranzato

Due to the variability and richness of natural data, it is almost impossible to build an accurate classifier by hand!

Deep Learning

Credit: Yishay

What society thinks I do



What my friends think I do

What other computer scientists think I do

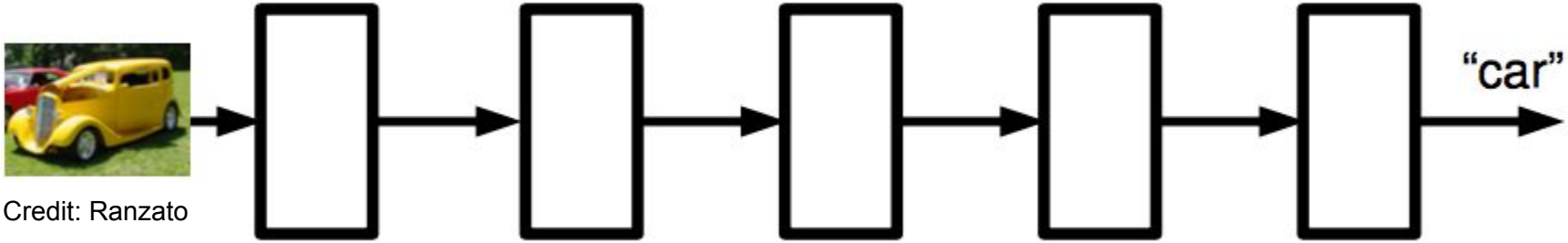
What mathematicians think I do

What I think I do

```
from theano import *
```

What I actually do

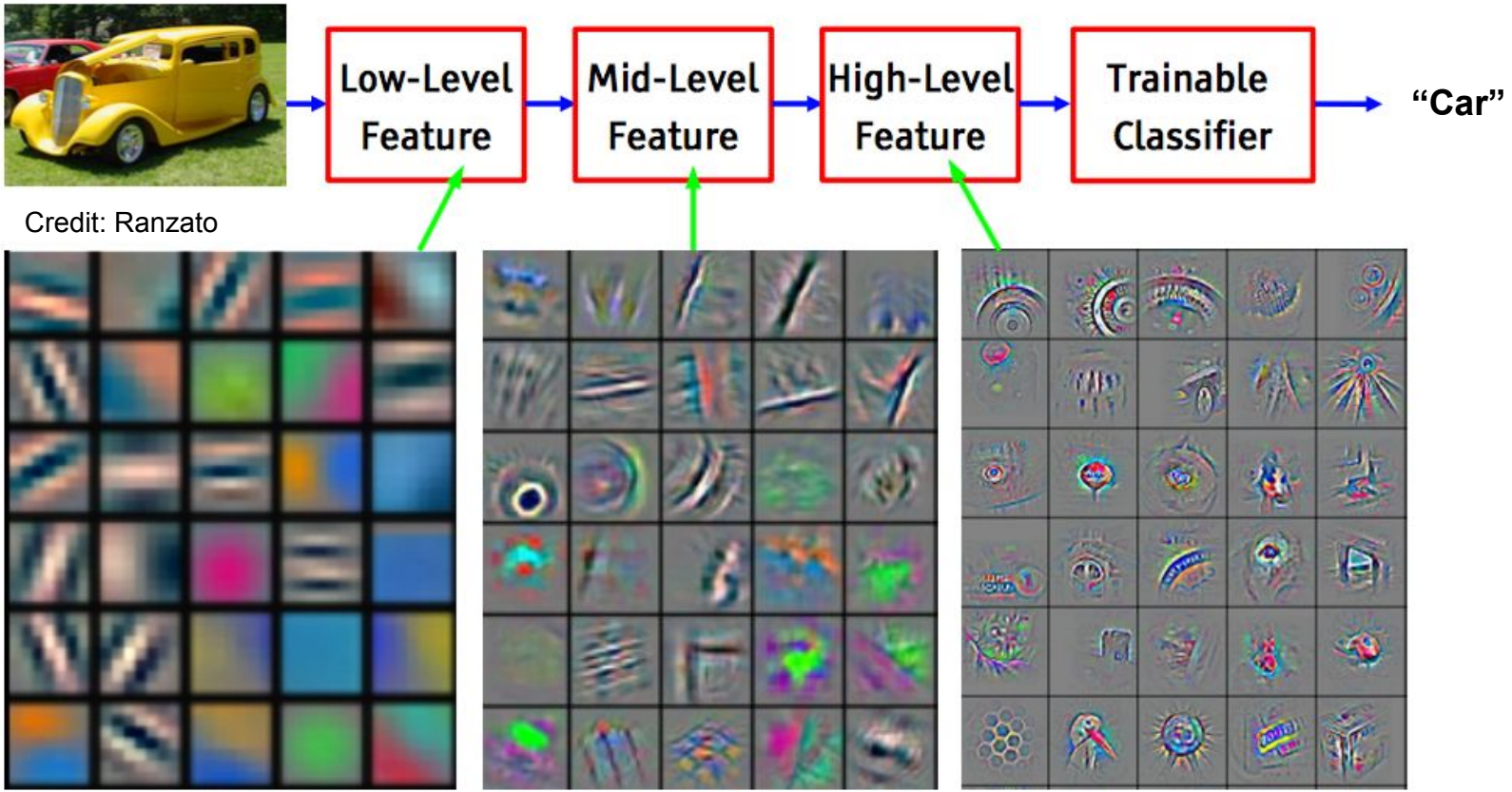
“Deep” Learning (Deep Neural Net)



Credit: Ranzato

- Artificial Neural Network that are composed of more than 1 hidden layer
- Depth of neural net allows it to construct a **Feature Hierarchy** of increasing abstraction.
- Each layer is a stage of non-linear transformation.

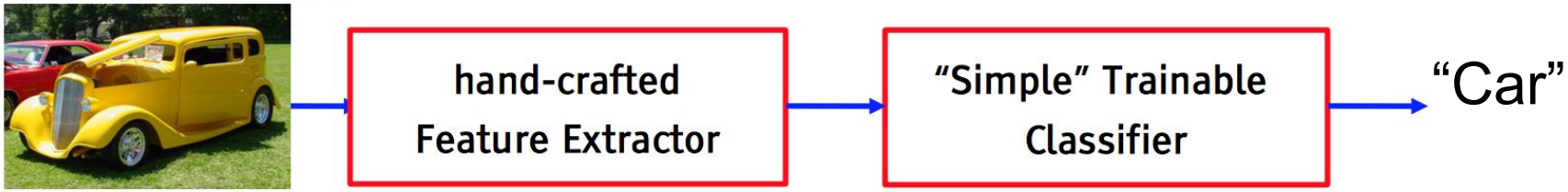
Feature Hierarchy: Simple to Abstract



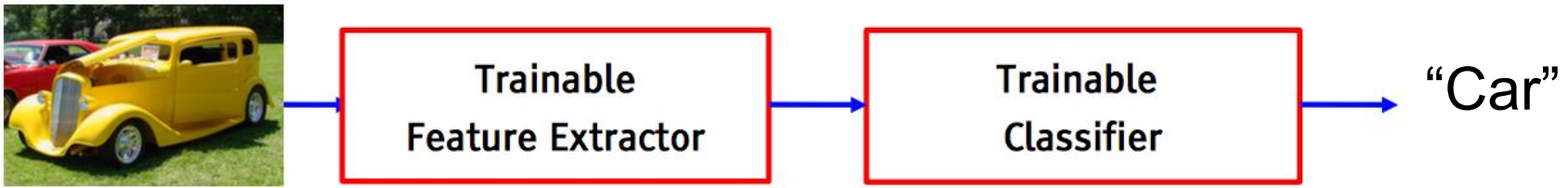
Credit: Ranzato

Traditional vs Deep Learning

- Traditional approach
  - Fixed/engineered features (or fixed kernel) + trainable classifier

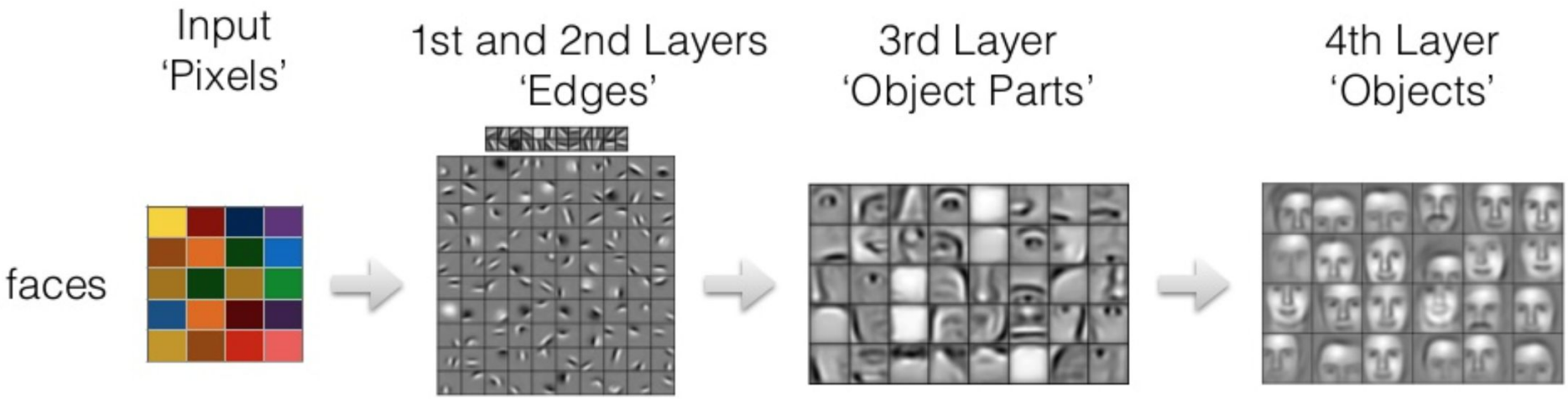


- End-to-end learning / Feature learning / Deep learning
  - Trainable features (or kernel) + trainable classifier



Credit: Ranzato

Example: Face Classifier



Credit: Caner

Visualizing the ‘black’ box

Credit: http://yosinski.com/deepvis

Deep Learning Methods

Unsupervised

- Restricted Boltzmann Machines
- Deep Belief Networks
- Auto encoders

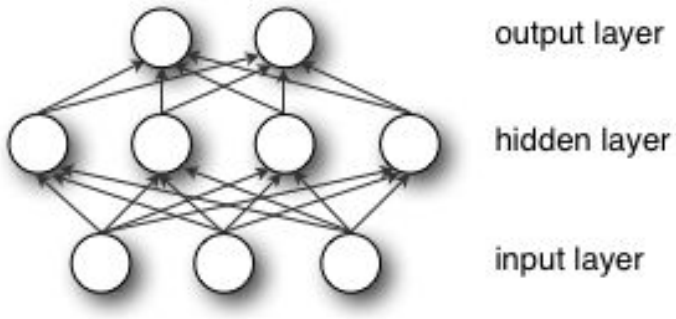
Supervised

- **Convolutional Neural Networks**
- Recurrent Neural Networks

Why Convolutional Net?

To understand why Convolutional Net is invented, let’s look at Multi-Layer Perceptron...

Multi-Layer Perceptron (MLP)

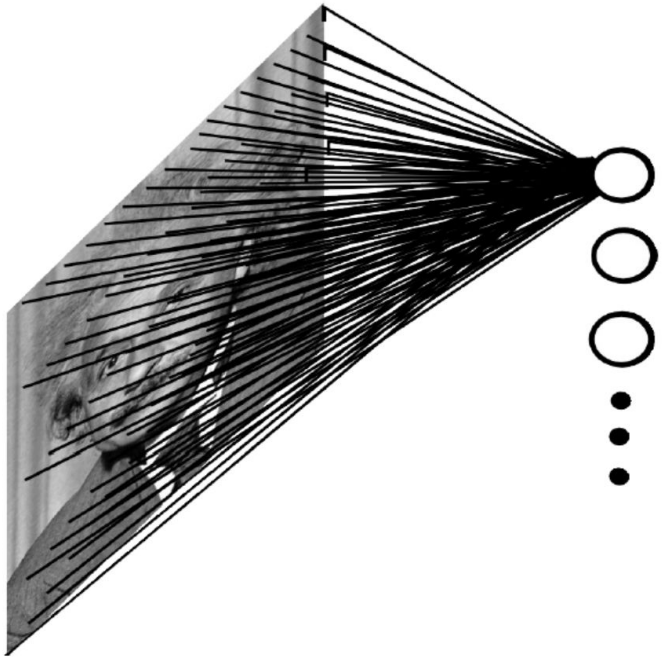


Credit: Theano

- Logistic regression classifier where its input is first transformed using a learnt non-linear transformation (hidden layer).
- A single hidden layer is sufficient to make MLP a **universal approximator**
- Why don’t we apply it to computer vision task?

What if the input is all the pixels within an image?

For a 200 x 200 image



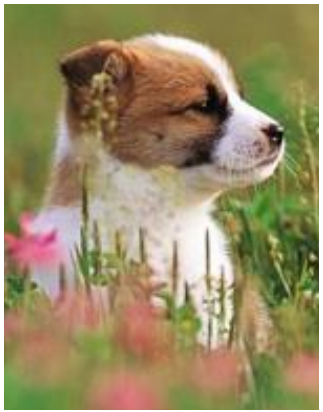
- 40,000 neurons each one with 40,000 inputs
- In total 1.6 Billions parameters!
- This is only for one single layer!!

Credit: Ranzato

Dog image classifier

Credit: Fanpop

Dog image classifier



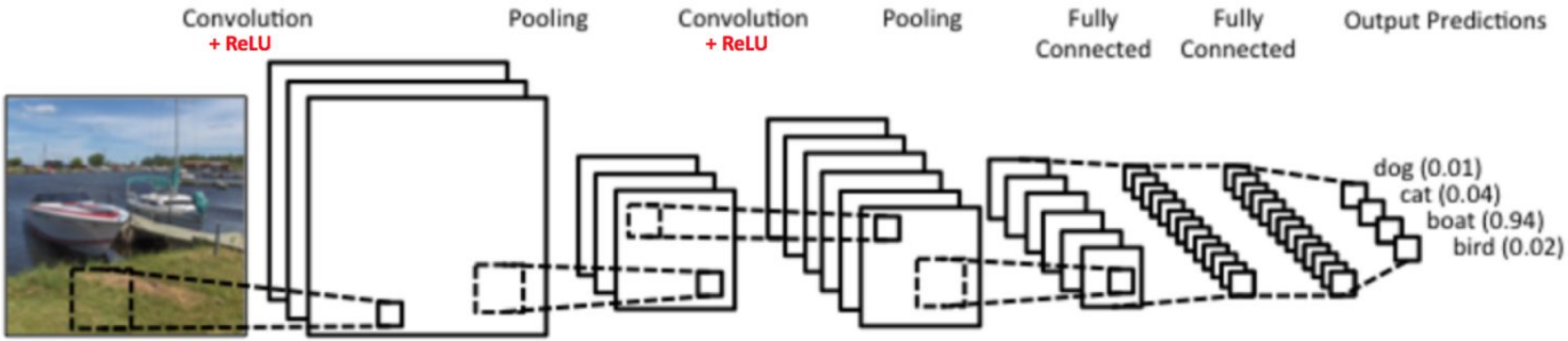
Observations

- Position doesn’t matter
  - Sharing weights between neurons (hidden units)
- Nearby pixels are more strongly correlated than distant pixels
  - Connect each hidden unit to a small patch of the input (local) instead of fully connected
  - Greatly reduce the no. of parameters

So, we achieve these using convolutional filters/kernels/neurons

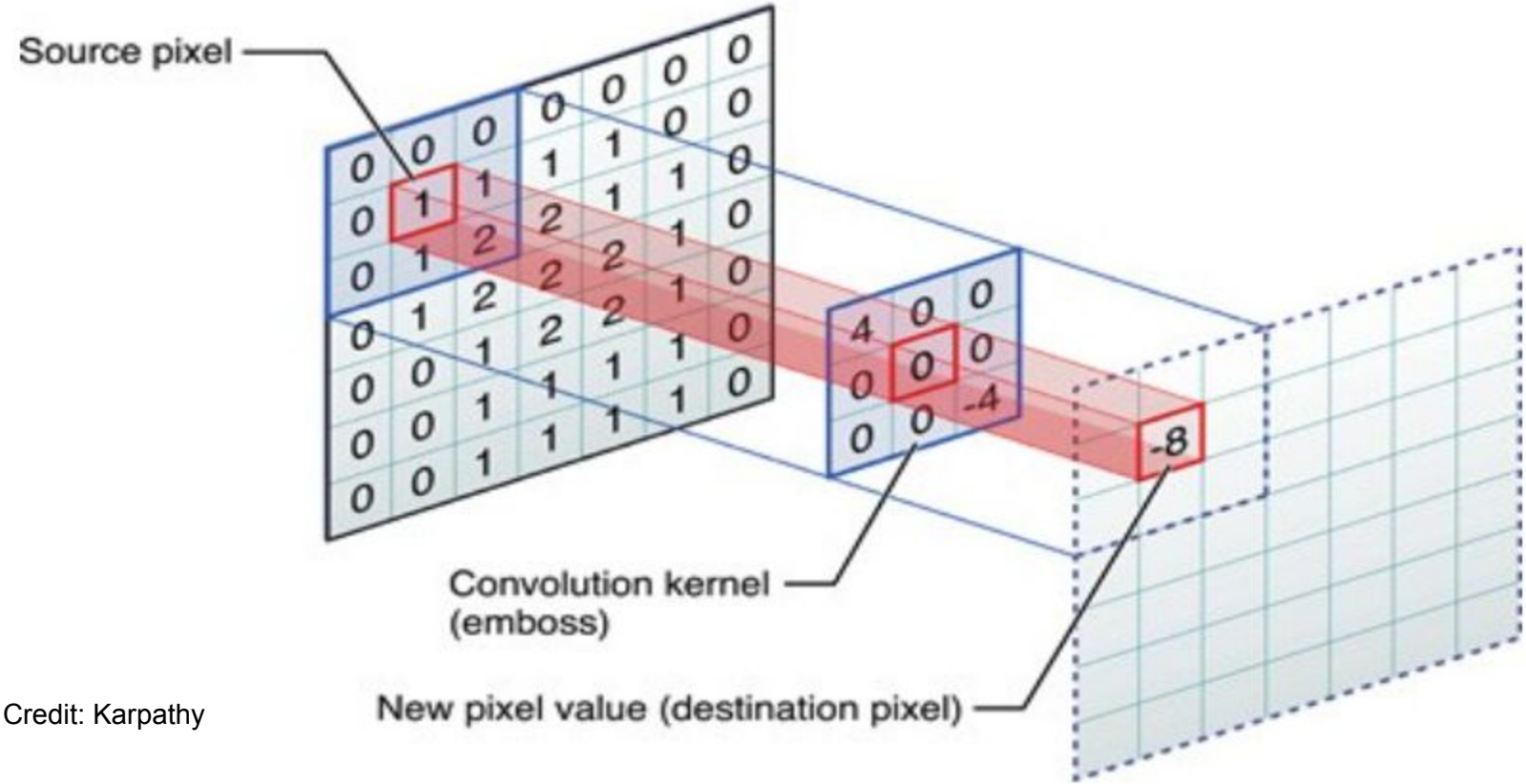
LeNet (Yann LeCun 1998)

- One of the very first convolutional neural networks which helped propel the field of Deep Learning.
- Convolution → Non-Linearity (ReLU) → Pooling (Max) → Fully Connected (MLP)



Credit: Yann LeCun

Convolution Filter / Kernel



Credit: Karpathy

Convolutional Filters/Kernels

Credit: GIMP

Blur

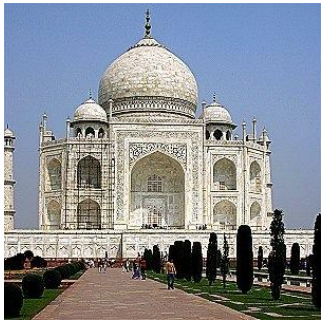
$$*\begin{bmatrix}0&0&0&0&0\\0&1&1&1&0\\0&1&1&1&0\\0&1&1&1&0\\0&0&0&0&0\end{bmatrix}=$$

Emboss

$$*\begin{bmatrix}&&&&\\&-2&-1&0&\\&-1&1&1&\\&0&1&2&\\&&&&\end{bmatrix}=$$

Sharpen

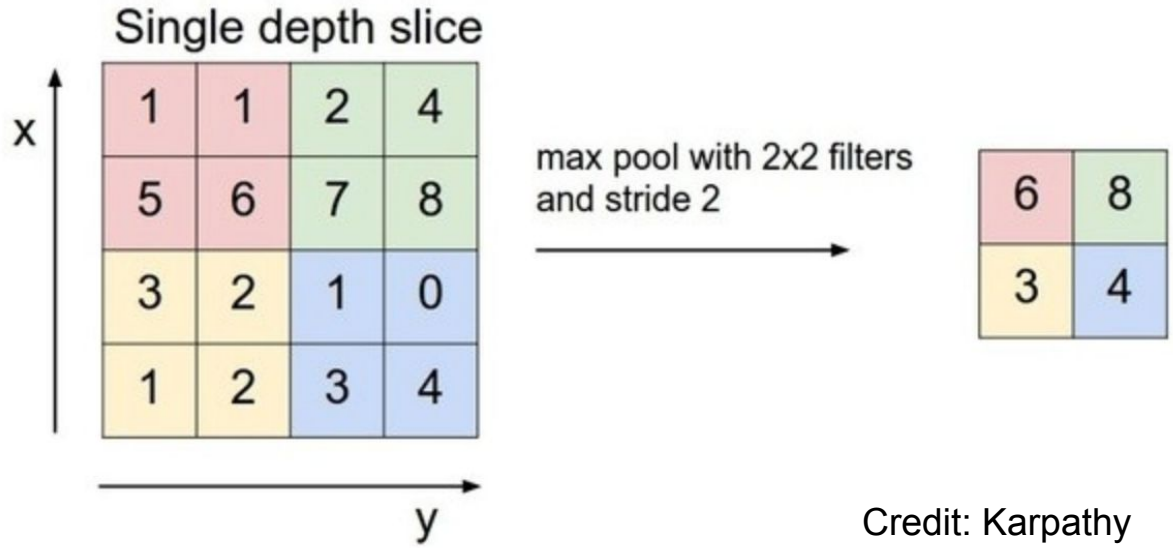
$$*\begin{bmatrix}0&0&0&0&0\\0&0&-1&0&0\\0&-1&5&-1&0\\0&0&-1&0&0\\0&0&0&0&0\end{bmatrix}=$$



Edge Detect

$$*\begin{bmatrix}&&&&\\&0&1&0&\\&1&-4&1&\\&0&1&0&\\&&&&\end{bmatrix}=$$

Max-pooling (Downsampling)



Credit: Karpathy

- Reduce no. of parameters → faster computation
- Reduce chances of overfitting

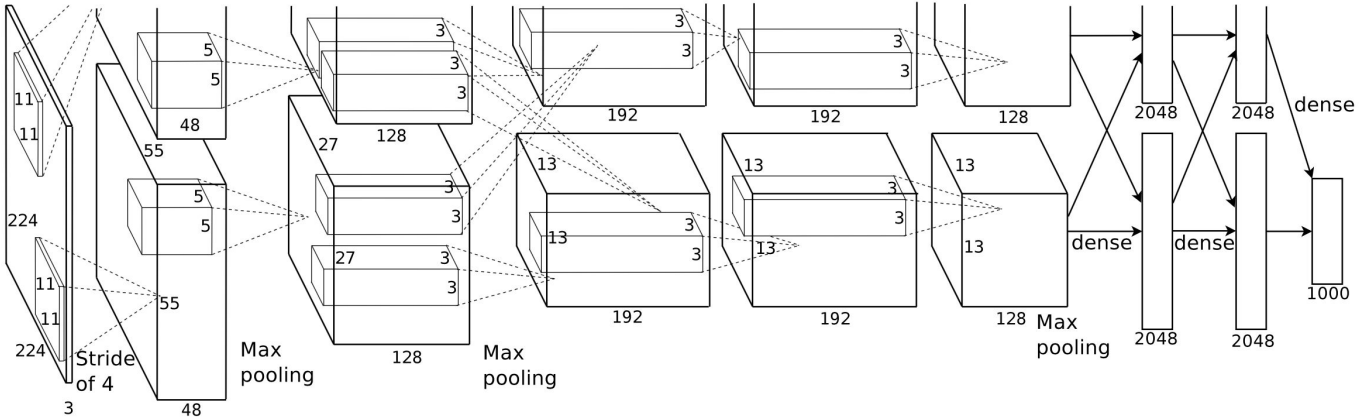
Why CNN works now?

- Faster parallel computing resources
  - CPU clusters, GPUs, etc
- Availability of large dataset
  - ImageNet: 1.2 millions images of 1,000 object classes
  - CoCo: 300k images of 2 millions object instances
- Improvements in model architectures

ConvNets Architectures

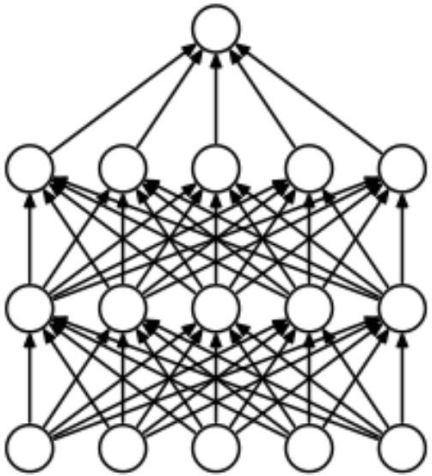
- AlexNet
- VGG
- Resnet

AlexNet (Winner of ILSVRC2012)

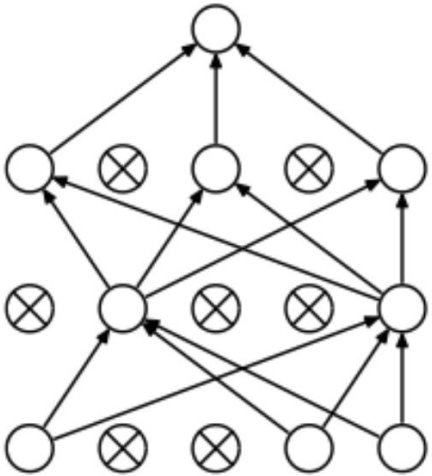


- A significant breakthrough, the paper was cited over 6,100 times!
- 8 layers (5 convolutional layers + 3 fully connected layers) with **Dropout**
- Took 5 ~ 6 days to train on 2 x GTX 580 3GB GPUs (1.2 million images)
- **ImageNet Classification with Deep Convolutional Neural Networks** [Alex et. al. 2012]

Dropout (Regularization)



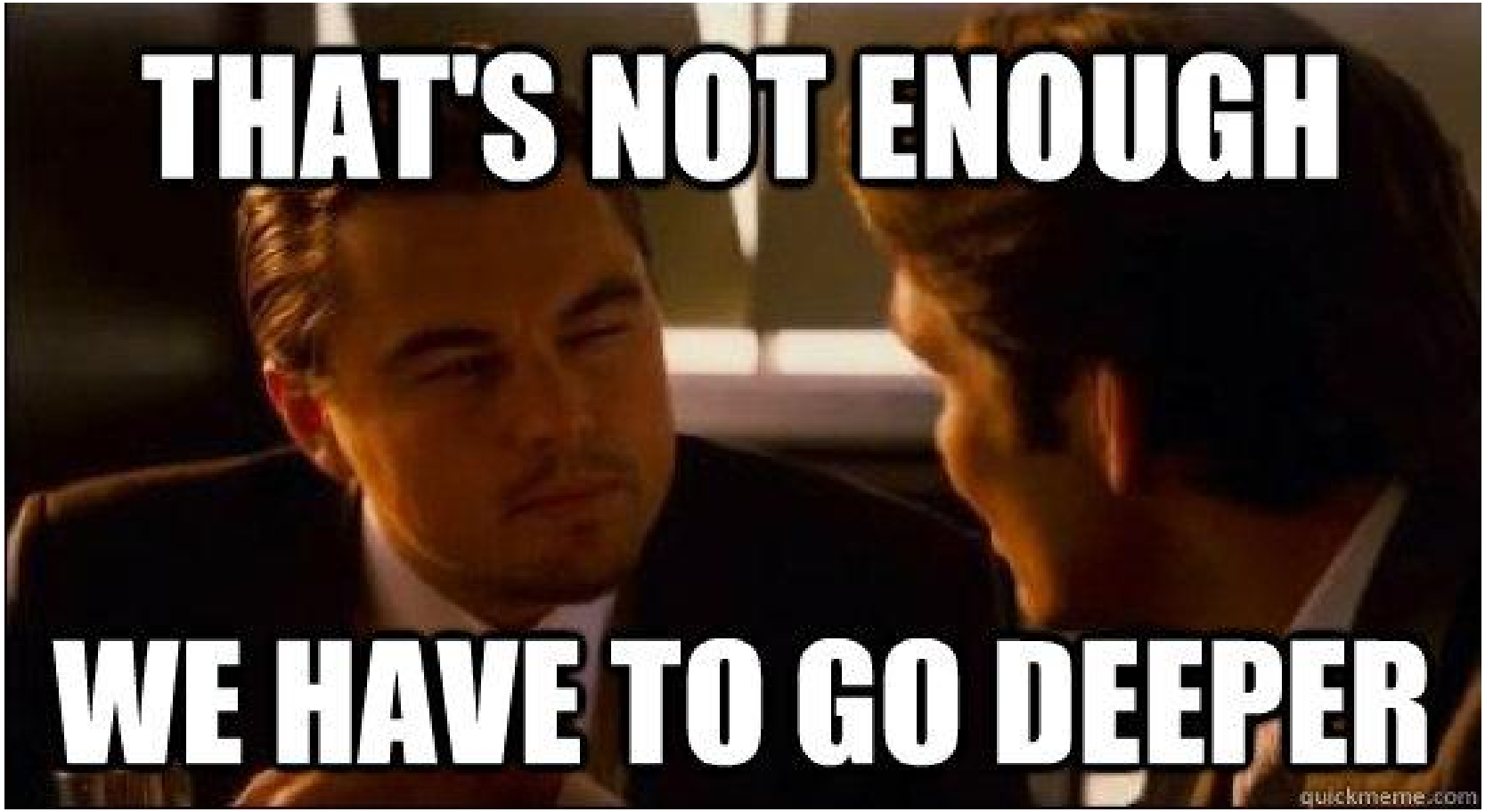
Standard Neural Net



After applying dropout.

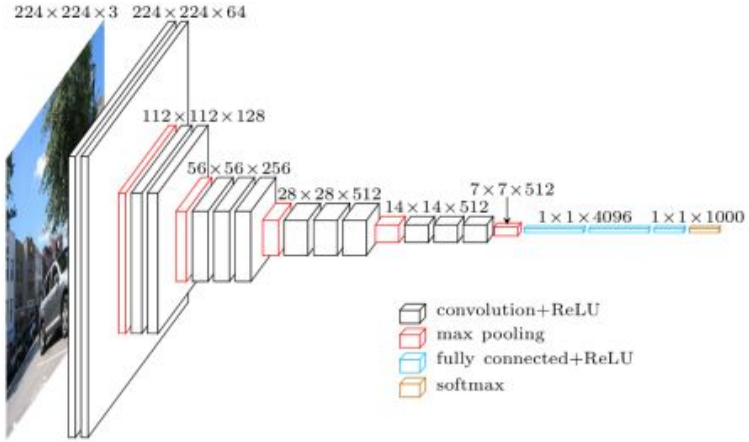
Credit: Hinton

- Reduce overfitting
- Randomly drop unit (with connection)
- Doubles the no. of iterations required to converge

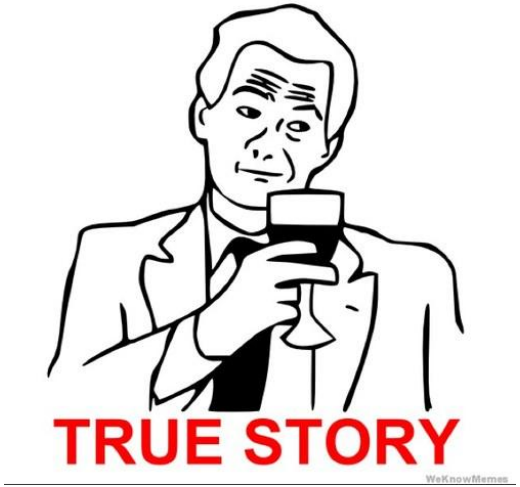

THAT'S NOT ENOUGH
WE HAVE TO GO DEEPER
quickmeme.com

VGG (Runner-up of ILSVRC2014)

- **Very Deep Convolutional Networks for Large-Scale Image Recognition** [Simonyan et al. 2015]
- Very small convolutional filters (**3 x 3**) in all layers
- Showed that the depth of network (**16 layers**) contributes to better performance
- Trained on 4 Nvidia Titan Black GPUs for **two to three weeks**

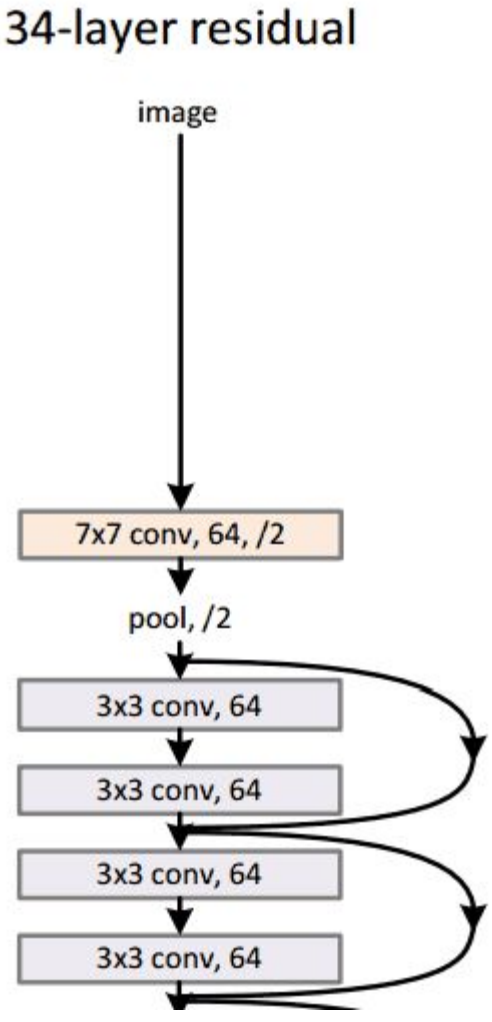



I WAS WINNING IMAGENET
UNTIL A DEEPER MODEL CAME ALONG
imgflip.com


TRUE STORY
WeKnowMemes

ResNet (Winner of ILSVRC & COCO 2015)

- **Deep Residual Learning for Image Recognition** [Kaiming He, et. al. 2015]
- 152 layers!
- Achieved 3.6% error rate against human (5%~ 10%)
- By adding shortcut connections that summed with the output of stacked convolutional layers
- "**Ultra-deep**" – Yann LeCun
- Trained on an 8 GPUs machine for **two to three weeks**



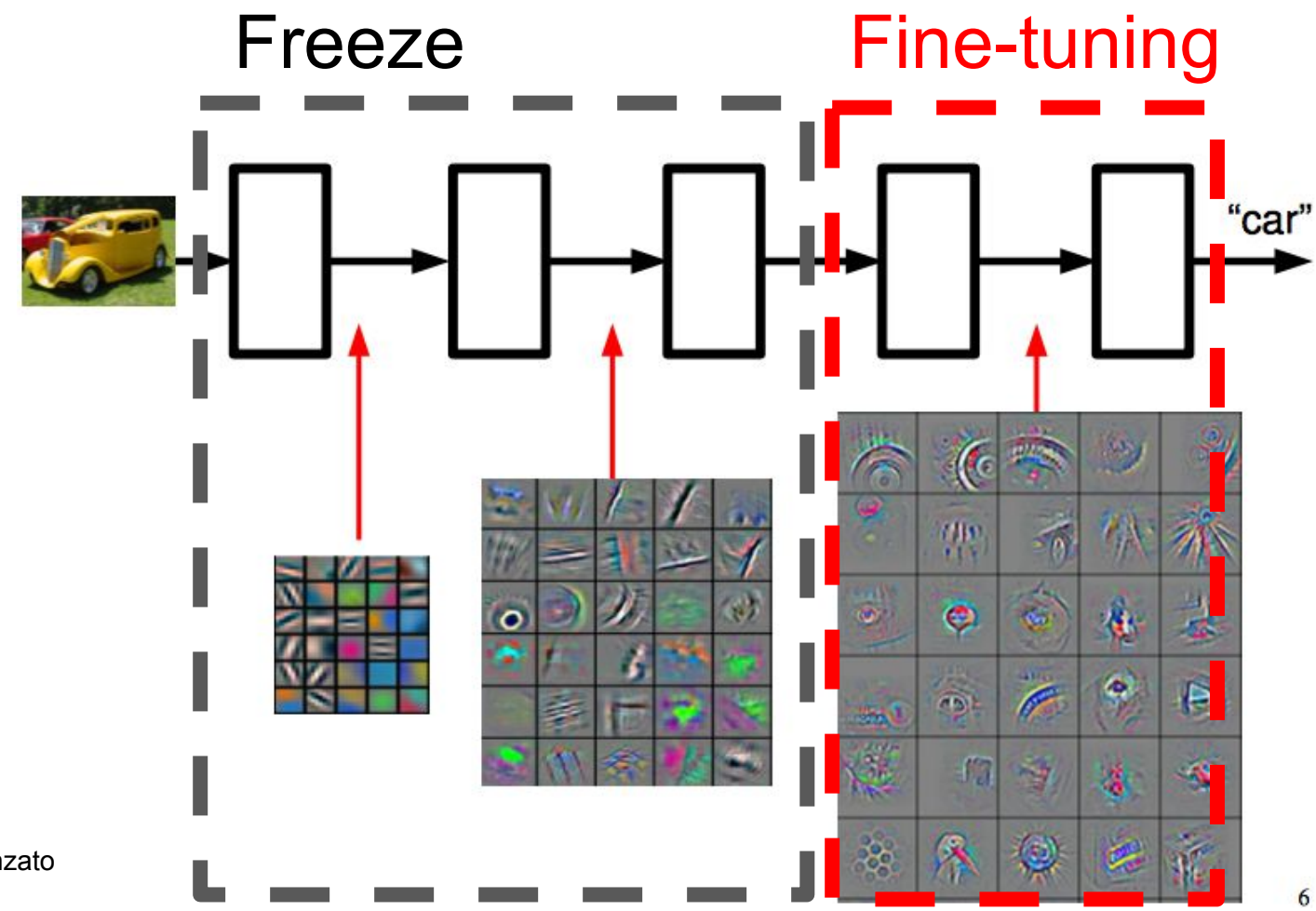
Credit: Adit Deshpande

How am I going to
train an effective ConvNets
without millions of images and
high-end GPUs?

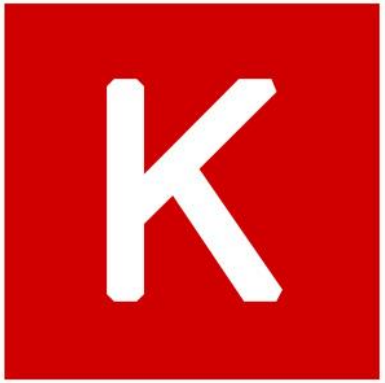
Transfer Learning!

Transfer Learning

1. Get a pre-trained model (weights & parameters)
    a. AlexNet, VGG, Resnet trained models are all available on the internet!
2. Replace the last layer (Fully Connected layer)
3. Freeze the weights of some layers (lower level)
4. Train model with your data!



Credit: Ranzato



dmlc
mxnet

Deep Learning Frameworks

theano

Apache SINGA

A General Distributed
Deep Learning Platform

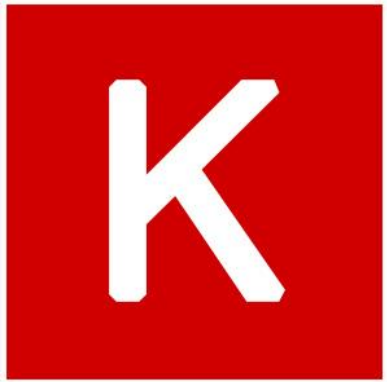
Software	Creator	Software license[a]	Open source	Platform	Written in	Interface	OpenMP support	OpenCL support	CUDA support	Automatic differentiation[1]	Has pretrained models	Recurrent nets	Convolutional nets	RBM/DBNs	Parallel execution (multi node)
Apache Singa	Apache Incubator	Apache 2.0	Yes	Linux, Mac OS X, Windows	C++	Python, C++, Java	No	Yes	Yes	?	Yes	Yes	Yes	Yes	Yes
Deeplearning4j	Skymind engineering team; Deeplearning4j community; originally Adam Gibson	Apache 2.0	Yes	Linux, Mac OS X, Windows, Android (Cross-platform)	C, C++	Java, Scala, Clojure, Python (Keras)	Yes	On roadmap[2]	Yes[3]	Computational Graph	Yes[4]	Yes	Yes	Yes	Yes[5]
Dlib	Davis King	Boost Software License	Yes	Cross-Platform	C++	C++	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes
Keras	François Chollet	MIT license	Yes	Linux, Mac OS X, Windows	Python	Python	Only if using Theano as backend	Under development for the Theano backend (and on roadmap for the TensorFlow backend)	Yes	Yes	Yes[6]	Yes	Yes	Yes	Yes[7]
Microsoft Cognitive Toolkit - CNTK	Microsoft Research	MIT license[8]	Yes	Windows, Linux[9] (OSX via Docker on roadmap)	C++	Python, C++, Command line,[10] BrainScript[11] (.NET on roadmap[12])	Yes[13]	No	Yes	Yes	Yes[14]	Yes[15]	Yes[15]	No[16]	Yes[17]
MXNet	Distributed (Deep) Machine Learning Community	Apache 2.0	Yes	Linux, Mac OS X, Windows,[18][19] AWS, Android,[20] iOS, JavaScript[21]	Small C++ core library	C++, Python, Julia, Matlab, JavaScript, Go, R, Scala, Perl	Yes	On roadmap[22]	Yes	Yes[23]	Yes[24]	Yes	Yes	Yes	Yes[25]
Neural Designer	Artelnics	Proprietary	No	Linux, Mac OS X, Windows	C++	Graphical user interface	Yes	No	No	?	?	No	No	No	?
OpenNN	Artelnics	GNU LGPL	Yes	Cross-platform	C++	C++	Yes	No	No	?	?	No	No	No	?
TensorFlow	Google Brain team	Apache 2.0	Yes	Linux, Mac OS X, Windows[26]	C++, Python	Python, (C/C++ public API only for executing graphs[27])	No	On roadmap[28][29]	Yes	Yes[30]	Yes[31]	Yes	Yes	Yes	Yes
Theano	Université de Montréal	BSD license	Yes	Cross-platform	Python	Python	Yes	Under development[32]	Yes	Yes[33][34]	Through Lasagne's model zoo[35]	Yes	Yes	Yes	Yes[36]
Torch	Ronan Collobert, Koray Kavukcuoglu, Clement Farabet	BSD license	Yes	Linux, Mac OS X, Windows,[37] Android,[38] iOS	C, Lua	Lua, LuaJIT,[39] C, utility library for C++/OpenCL[40]	Yes	Third party implementations[41][42]	Yes[43][44]	Through Twitter's Autograd[45]	Yes[46]	Yes	Yes	Yes	Yes[47]
Wolfram Mathematica	Wolfram Research	Proprietary	No	Windows, Mac OS X, Linux, Cloud computing	C++	Command line, Java, C++	No	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes

Source: https://en.wikipedia.org/wiki/Comparison_of_deep_learning_software

Factors to consider

- Modeling capability
  - Ability to train common and state-of-the-art networks without writing too much code. Also, the flexibility to create a new architecture.
- Interface
  - Python, Java, Lua, etc.
- Model deployment
  - How easy to deploy or productionize.
- Performance:
  - Single/Multi GPUs
- Community
  - Widely used / backed by big tech co.

Beginner + Python =




תודה
Dankie
Gracias
Спасибо
شكراً
Merci
Takk
Köszönjük
Terima kasih
Grazie
Dziękujemy
Dėkojame
Ďakujeme
Vielen Dank
Paldies
Kiitos
Täname teid
谢谢
Thank You
Tak
感謝您
Obrigado
Teşekkür Ederiz
감사합니다
Σας Ευχαριστούμ
ขอบคุณ
Bedankt
Děkujeme vám
ありがとうございます
Tack

Appendix

Tips on training a deep neural net

http://rishy.github.io/ml/2017/01/05/how-to-train-your-dnn/