Convolutional Neural Networks

For Image Classification

Cape Town Deep Learning Meet-up 20 June 2017

Alex Conway
alex @ numberboost.com
NUMBERBOOST

https://github.com/alexcnwy/CTDL_CNN_TALK_20170620

Hands up!

Big Shout Outs

Jeremy Howard & Rachel Thomas

http://course.fast.ai

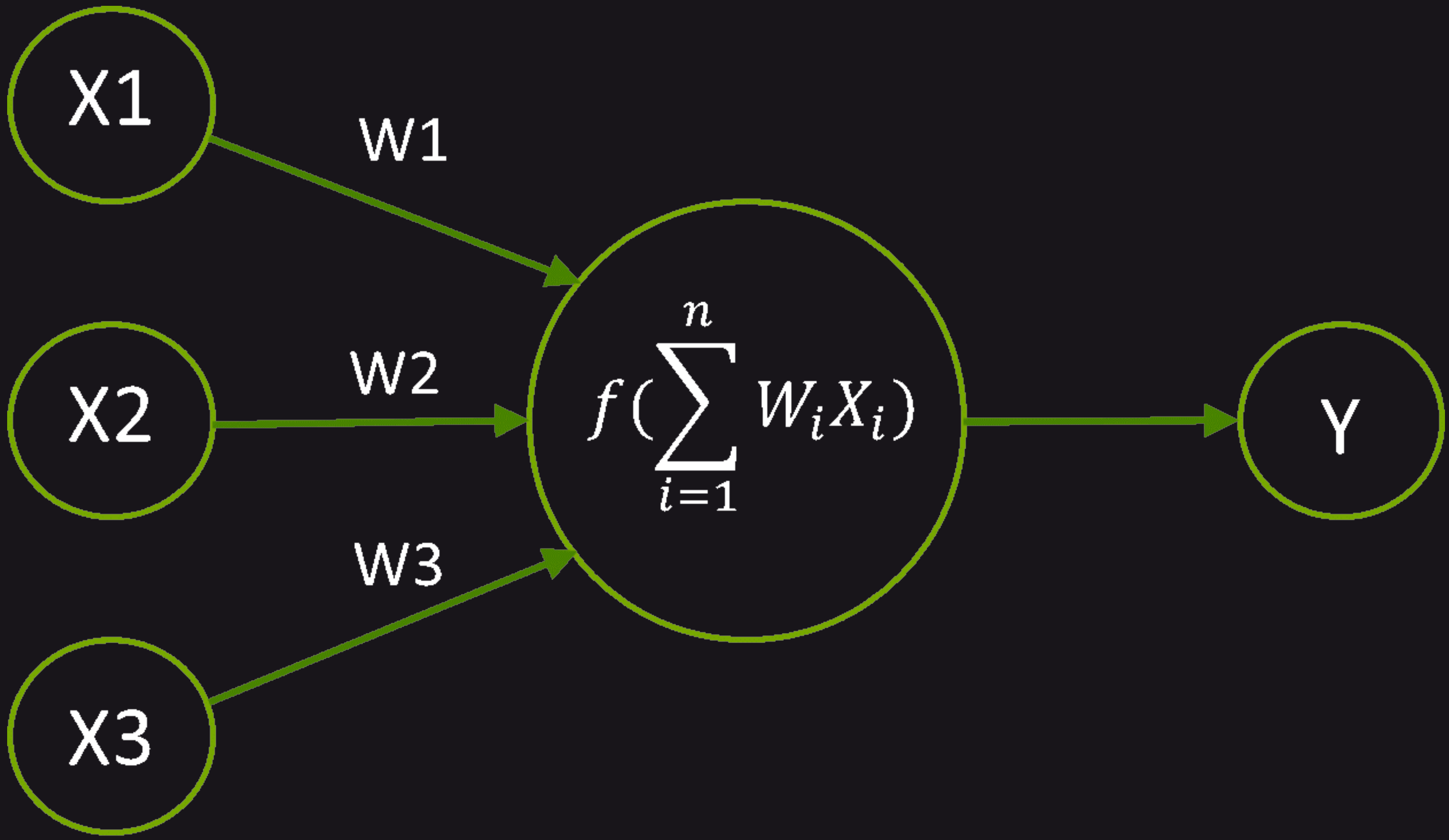
Andrej Karpathy

http://cs231n.github.io

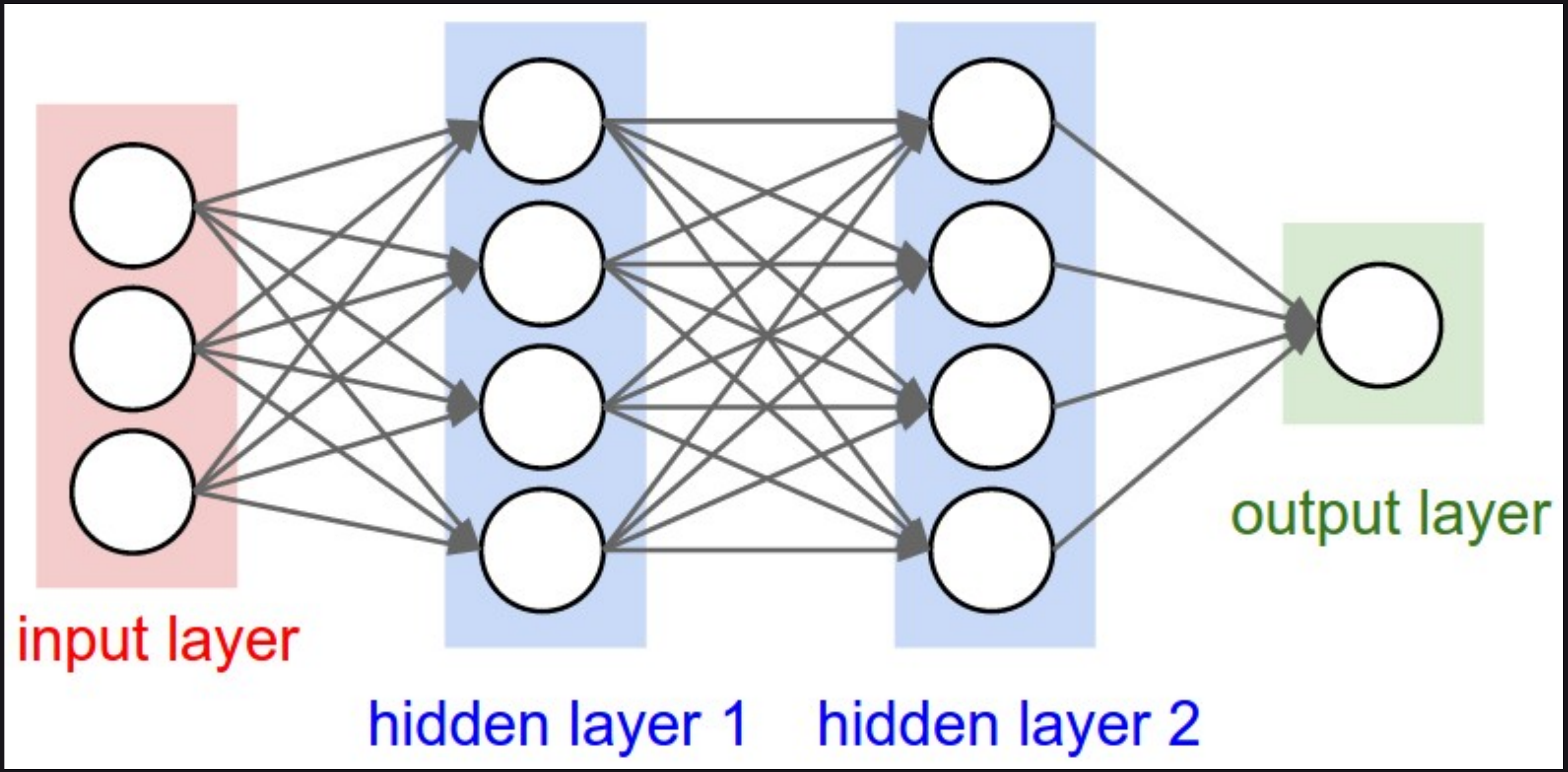
1. What is a neural network?
2. What is an image?
3. What is a convolutional neural network?
4. Using a pre-trained ImageNet-winning CNN
5. Fine-tuning a CNN to solve a new problem
6. Visual similarity “latest AI technology” app
7. Practical tips
8. Image cropping
9. Image captioning
10. CNN + Word2Vec
11. Style transfer
12. Where to from here?

1. What is a neural network?
2. What is an image?
3. What is a convolutional neural network?
4. Using a pre-trained ImageNet-winning CNN
5. Fine-tuning a CNN to solve a new problem
6. Visual similarity “latest AI technology” app
7. Practical tips
8. Image cropping
9. Image captioning
10. CNN + Word2Vec
11. Style transfer
12. Where to from here?

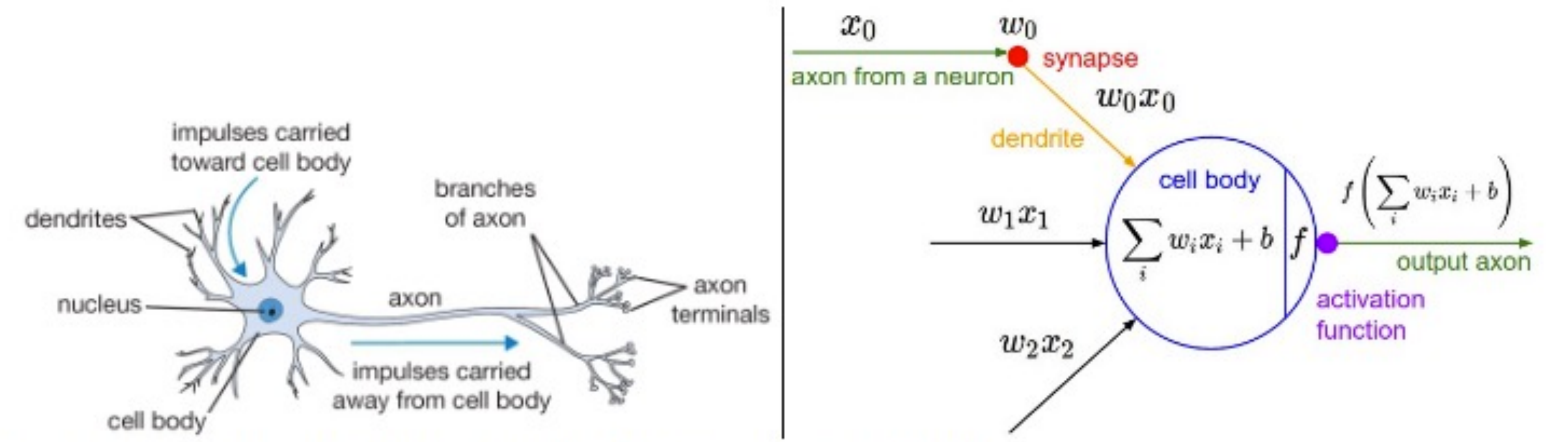
What is a Neural Network?



What is a Neural Network?



What is a Neural Network?



A cartoon drawing of a biological neuron (left) and its mathematical model (right).

http://playground.tensorflow.org

What is a Neural Network?

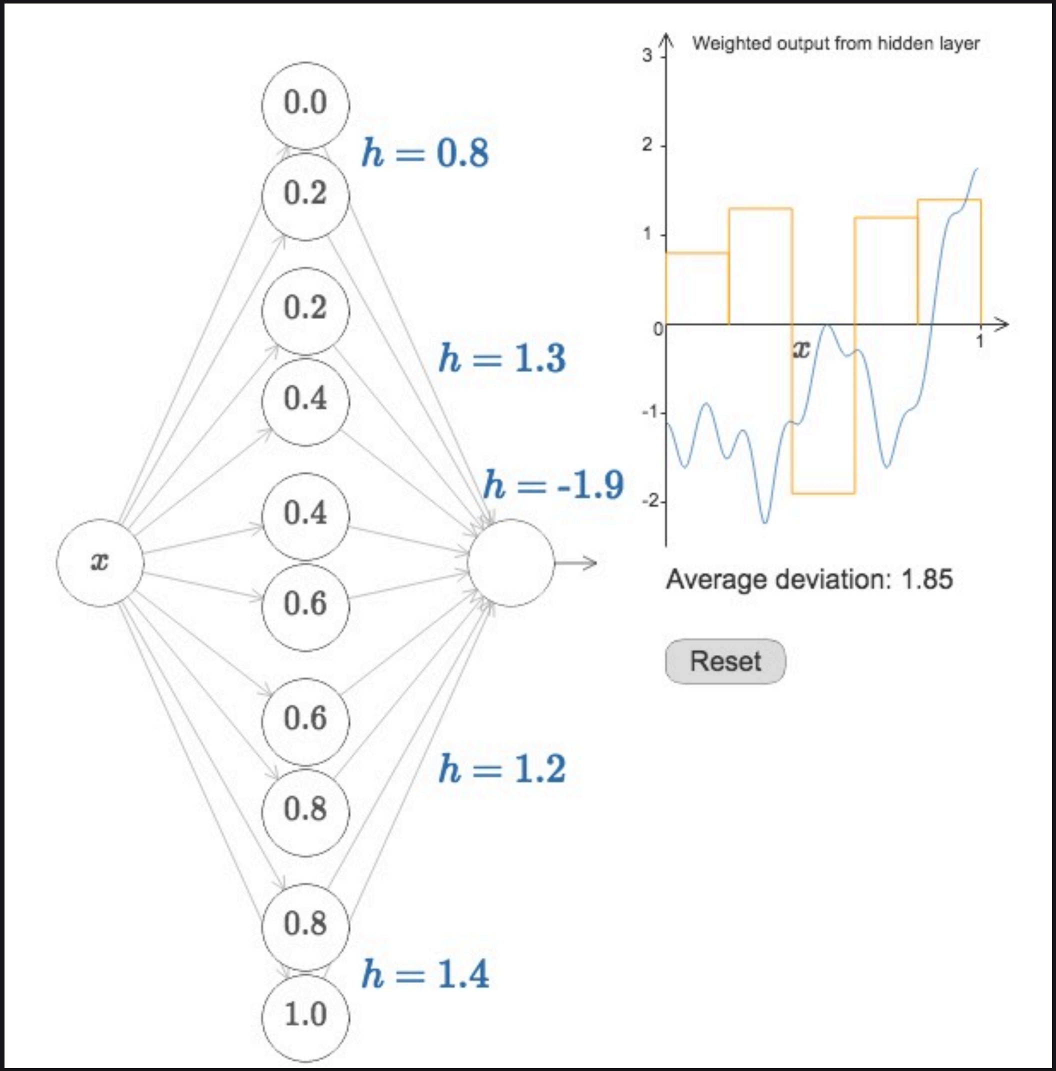
For much more detail, see:

1. *Michael Nielson's Neural Networks & Deep Learning free online book*
http://neuralnetworksanddeeplearning.com/chap1.html

2. *Anrej Karpathy's CS231n Notes*
http://neuralnetworksanddeeplearning.com/chap1.html

What is a Neural Network?

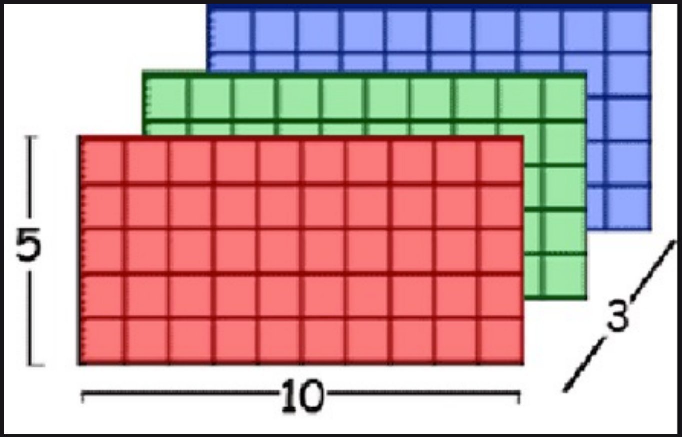
Universal Approximation theorem:



http://neuralnetworksanddeeplearning.com/chap4.html

What is an Image?

- Pixel = 3 colour channels (R, G, B)
- Pixel intensity = number in [0,255]
- Image has width w and height h
- Therefore image is w x h x 3 numbers

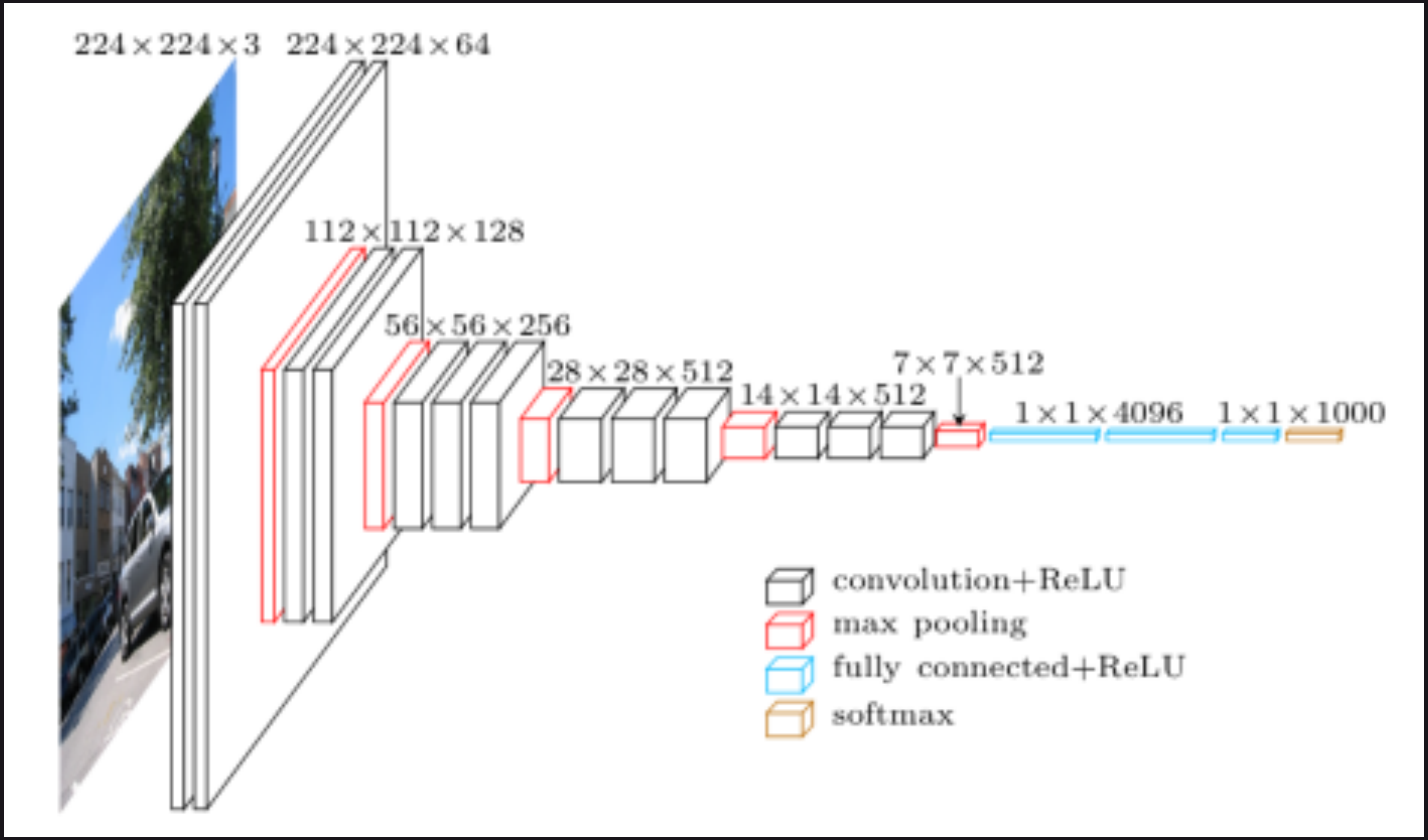


What is a Convolutional Neural Network (CNN)?

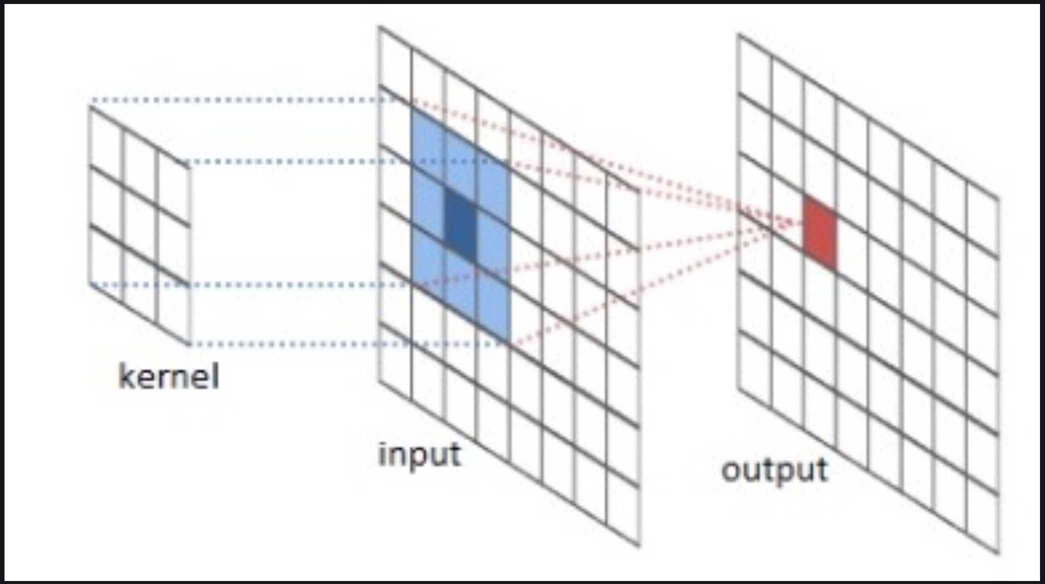
CNN = Neural Network + Image

- with some tricks -

What is a Convolutional Neural Network (CNN)?



Convolutions



- 2-d weighted average
- Element-wise multiply kernel with pixels
- “learn” the kernels
- http://setosa.io/ev/image-kernels/
- http://cs231n.github.io/convolutional-networks/

Convolutions

“imagine taking this 3x3 matrix (“kernel”) and positioning it over a 3x3 area of an image, and let's multiply each overlapping value. Next, let's sum up these products, and let's replace the center pixel with this new value. If we slide this 3x3 matrix over the entire image, we can construct a new image by replacing each pixel in the same manner just described.”

Convolutions

"...we understand that filters can be used to identify particular visual "elements" of an image, it's easy to see why they're used in deep learning for image recognition. But how do we decide which kinds of filters are the most effective? Specifically, what filters are best at capturing the necessary detail from our image to classify it?

- …these filters are just matrices that we are applying to our input to achieve a desired output... therefore, given labelled input, we don't need to manually decide what filters work best at classifying our images, we can simply train a model to do so, using these filters as weights!

Convolutions

“ ...for example, we can start with 8 randomly generated filters; that is 8 3x3 matrices with random elements. Given labeled inputs, we can then use stochastic gradient descent to determine what the optimal values of these filters are, and therefore we allow the neural network to learn what things are most important to detect in classifying images. “

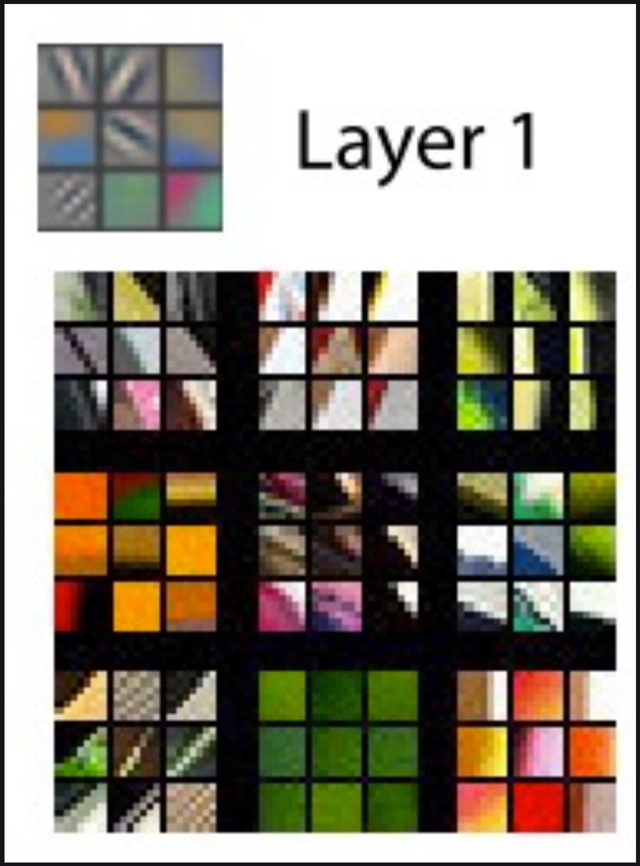
Convolutions

Visualizing and Understanding Convolutional Networks

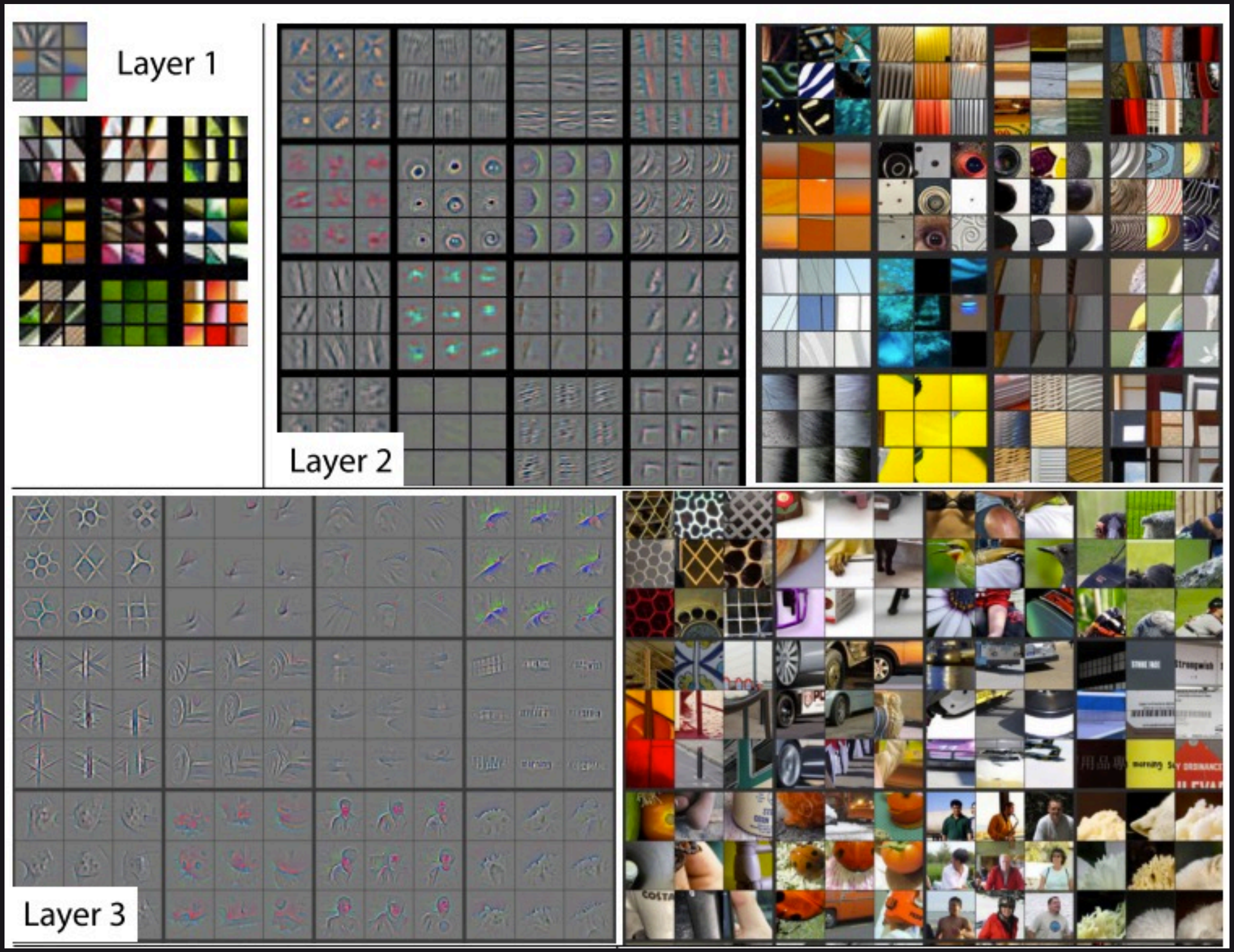
Matthew D. Zeiler ZEILER@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University

Rob Fergus FERGUS@CS.NYU.EDU
Dept. of Computer Science, Courant Institute, New York University

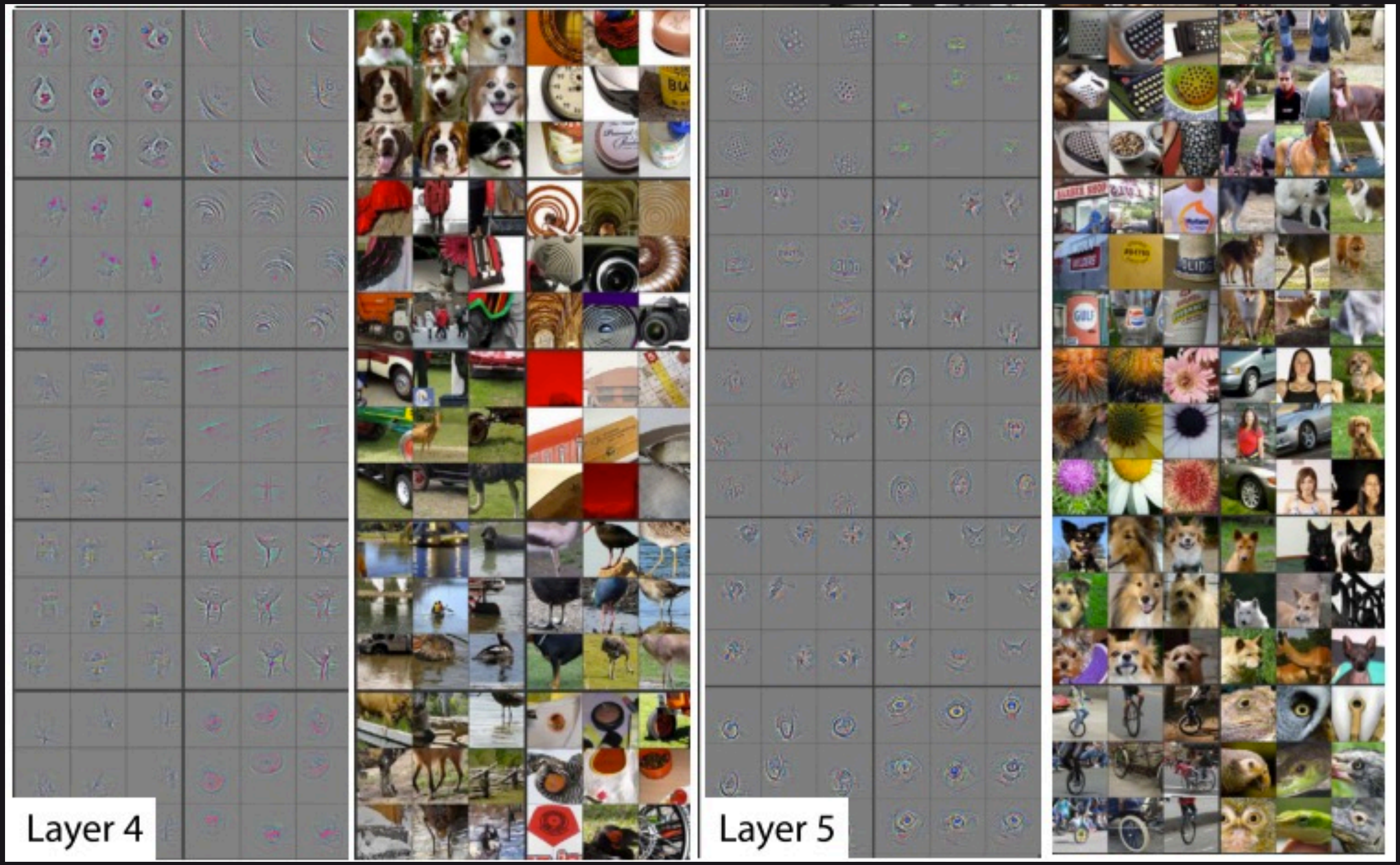
Convolutions



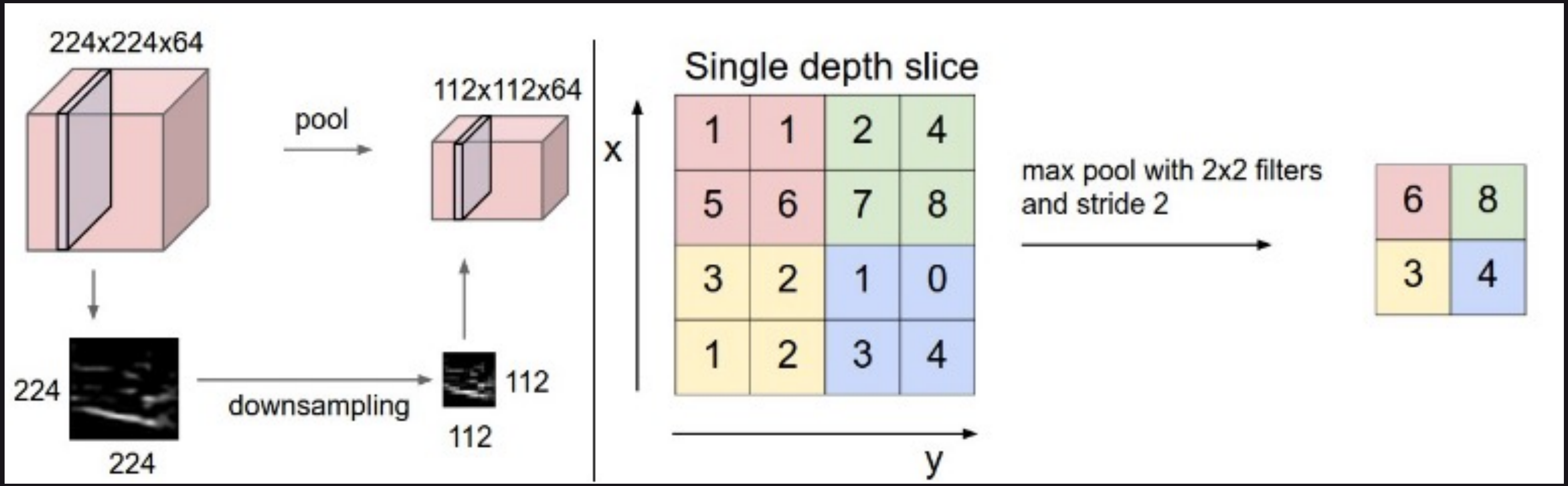
Convolutions



Convolutions

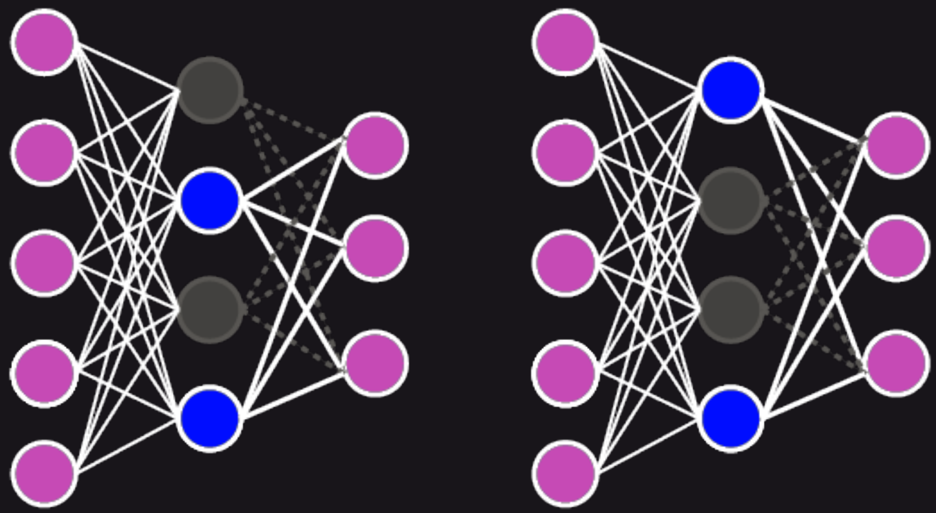


Max Pooling



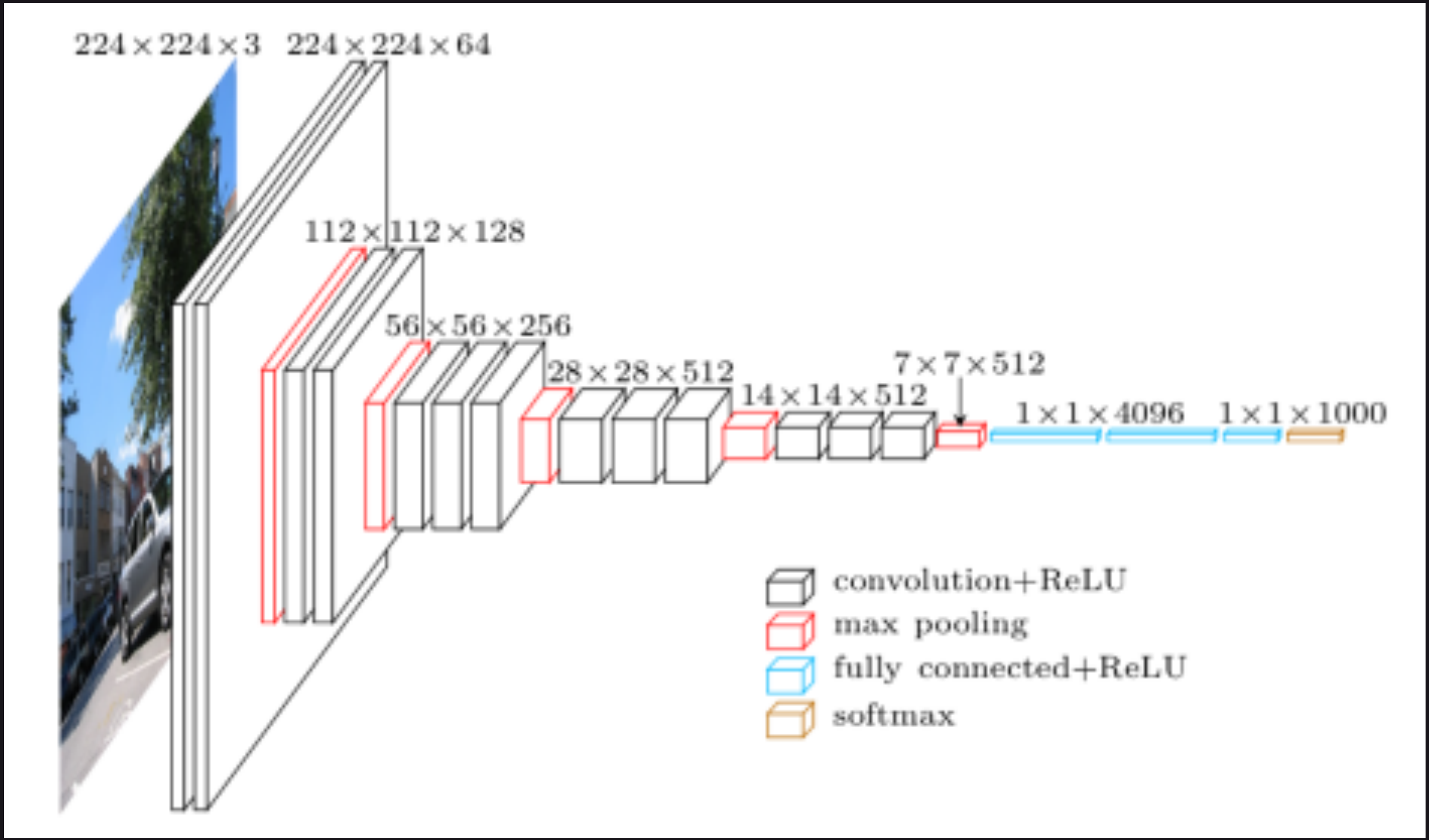
- Reduces dimensionality from one layer to next
- By replacing NxN sub-area with max value
- Makes network "look" at larger areas of the image at a time e.g. Instead of identifying fur, identify cat
- Reduces computational load
- Controls for overfitting

Dropout



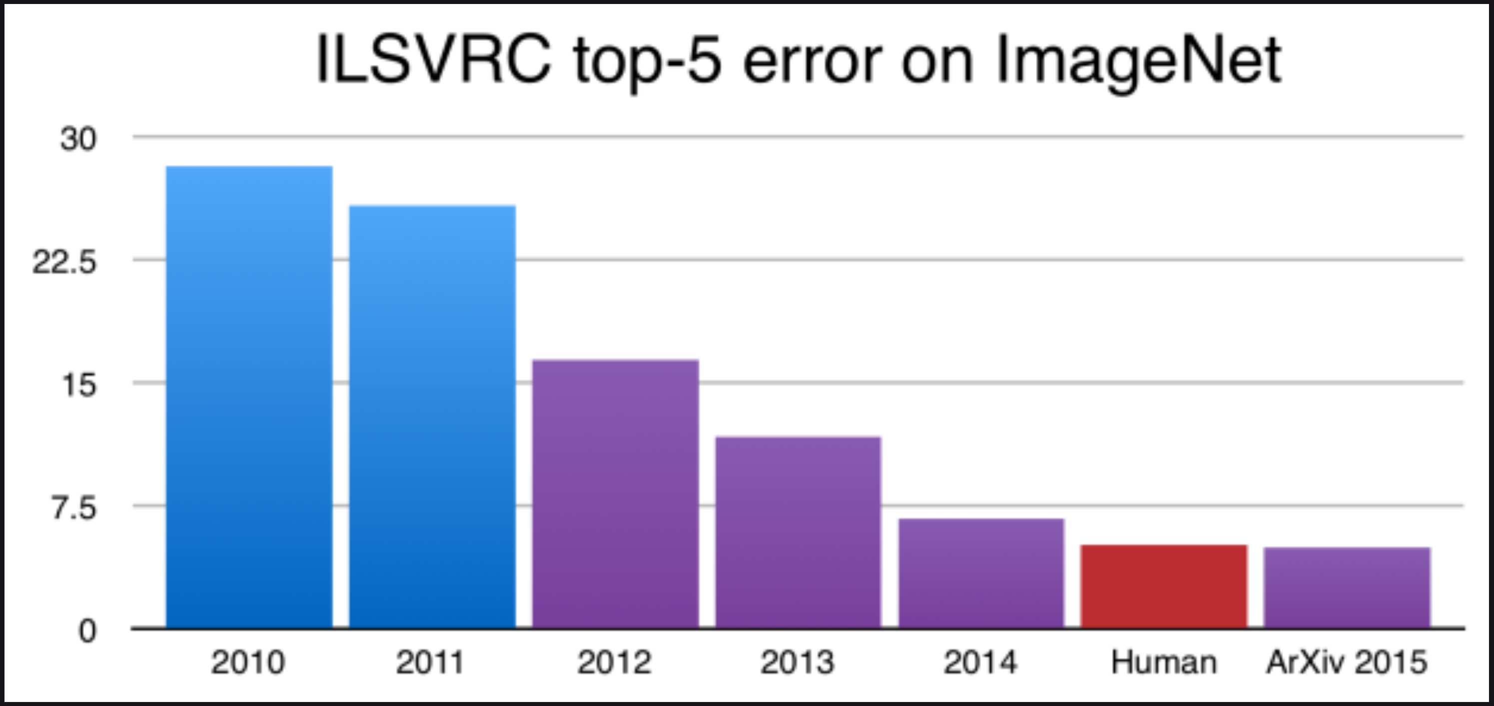
- Form of regularization (helps prevent overfitting)
- Trades ability to fit training data to help generalize to new data
- Used during training (not test)
- Randomly set weights in hidden layers to 0 with some probability p

CNN Architectures



Using a Pre-Trained ImageNet-Winning CNN

http://image-net.org/explore



Using a Pre-Trained ImageNet-Winning CNN

Using a Pre-Trained ImageNet-Winning CNN

- We'll be using "VGGNet"
- Oxford Visual Geometry Group (VGG)
- The runner-up in ILSVRC 2014
- Network contains 16 CONV/FC layers (deep!)
- The whole VGGNet is composed of CONV layers that perform 3x3 convolutions with stride 1 and pad 1, and of POOL layers that perform 2x2 max pooling with stride 2 (and no padding)
- Its main contribution was in showing that the depth of the network is a critical component for good performance.
- Homogeneous architecture that only performs 3x3 convolutions and 2x2 pooling from the beginning to the end.
- Easy to fine-tune

Using a Pre-Trained ImageNet-Winning CNN

Published as a conference paper at ICLR 2015

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

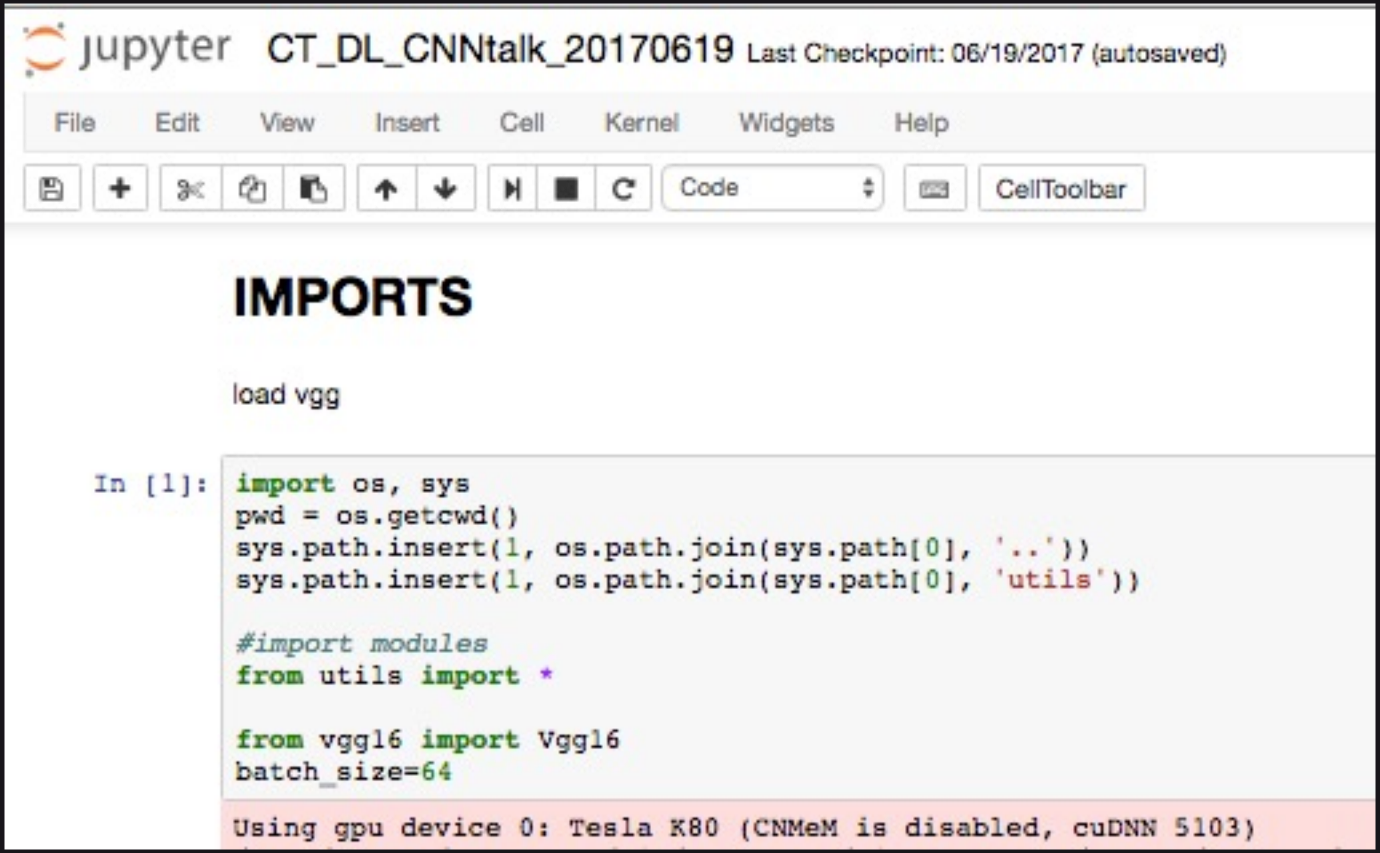
Karen Simonyan* & **Andrew Zisserman**+
Visual Geometry Group, Department of Engineering Science, University of Oxford
{karen,az}@robots.ox.ac.uk

Using a Pre-Trained ImageNet-Winning CNN

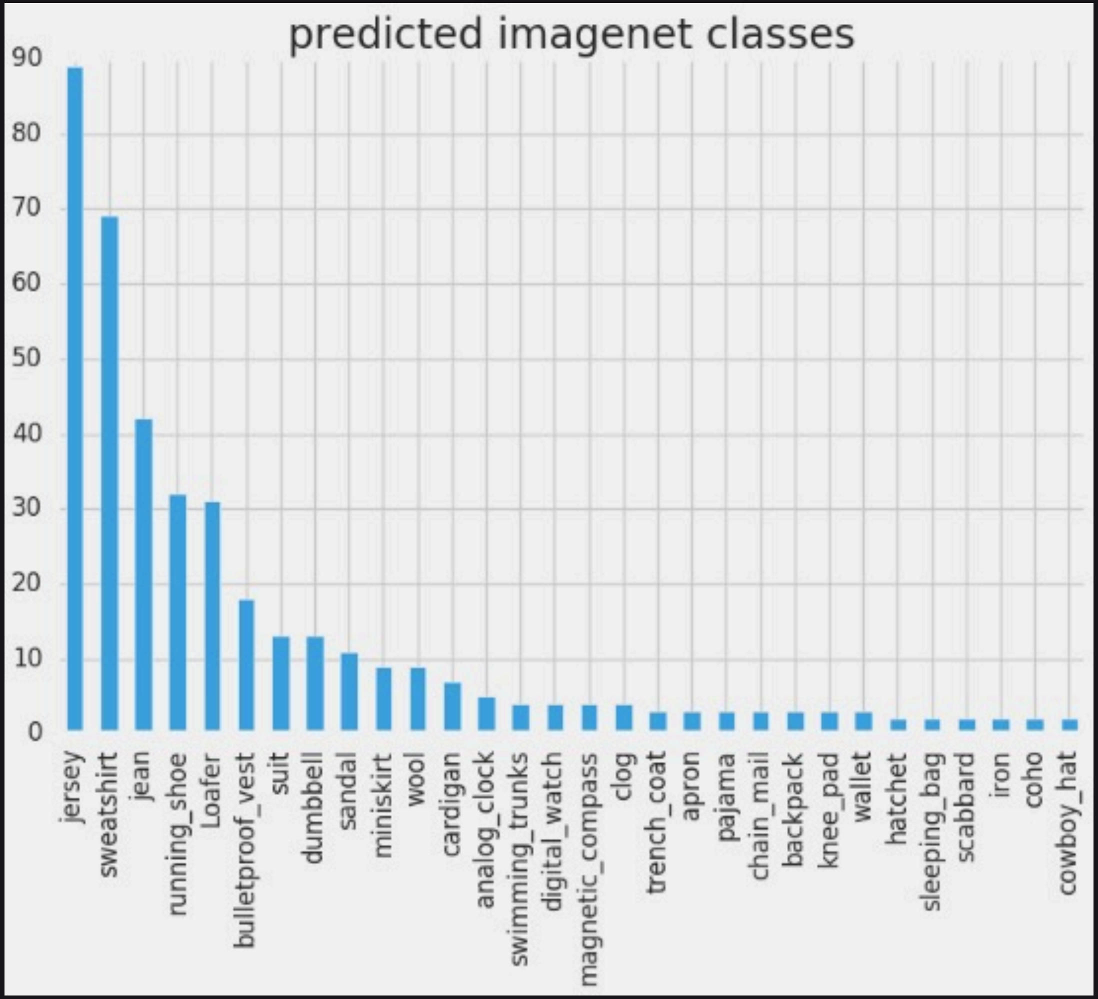
CODE TIME!

https://github.com/alexcnwy/CTDL_CNN_TALK_20170620

https://github.com/alexcnwy/CTDL_CNN_TALK_20170620



https://github.com/alexcnwy/CTDL_CNN_TALK_20170620



Fine-tuning A CNN To Solve A New Problem

- Fix weights in convolutional layers (trainable=False)
- Re-train final dense layer(s)

```
# grab vgg keras model object
m = vgg.model
```

```
m.summary()
```

...

```
# chop off last 2 layers
m.pop()
m.pop()
```

```
# now final layer is:
# dense_2 (Dense)                (None, 4096)          0            dropout_1[0][0]
m.summary()
```

Visual Similarity “Latest AI Technology” App

https://memeburn.com/2017/06/spree-image-search/

Visual Similarity “Latest AI Technology” App

- Chop off last 2 layers
- Use dense layer with 4096 activations
- Compute nearest neighbours in the space of these activations

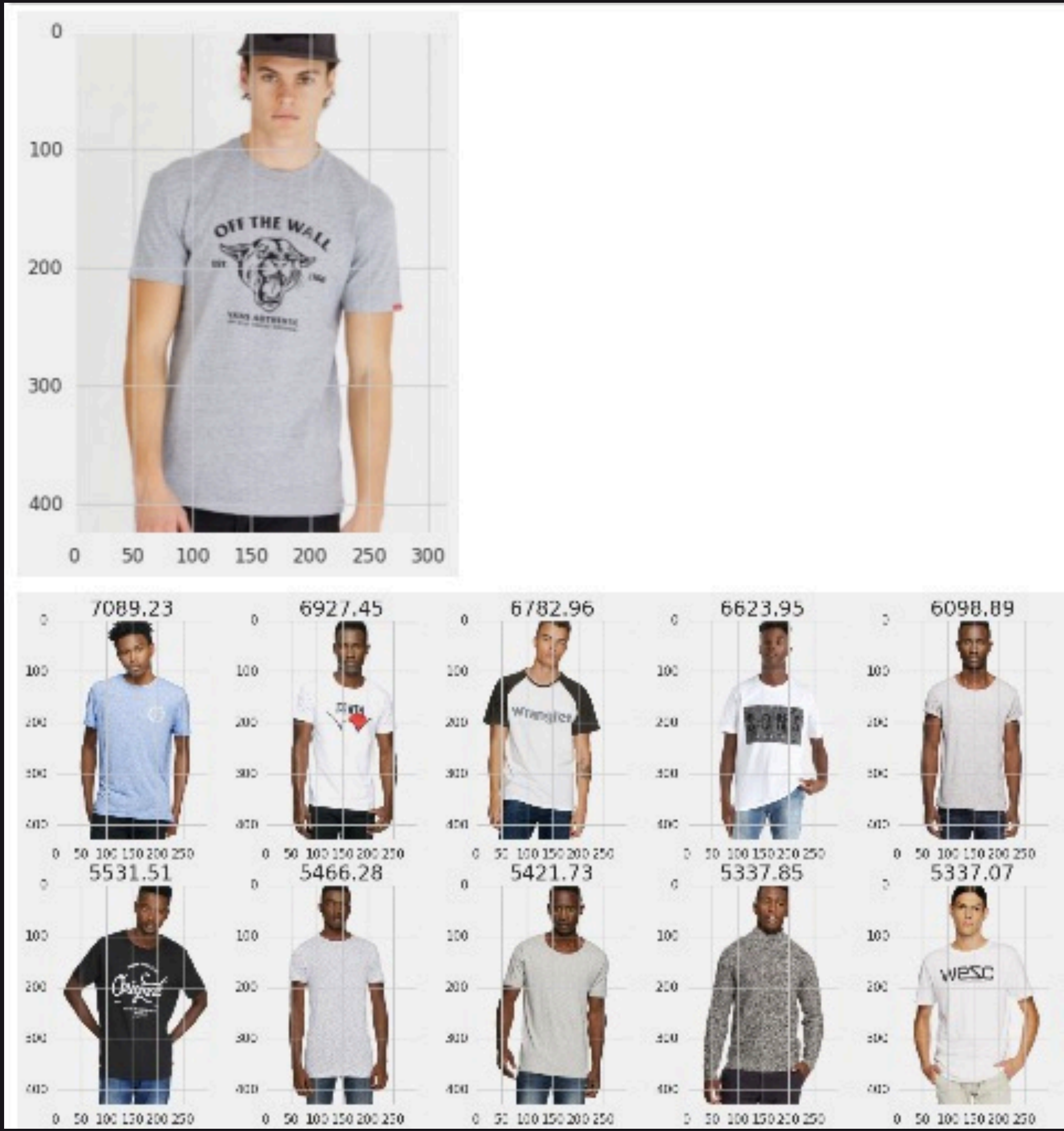
CODE TIME!

https://github.com/alexcnwy/CTDL_CNN_TALK_20170620

https://github.com/alexcnwy/CTDL_CNN_TALK_20170620

```python
def get_most_similar_products(test_img_idx):
    # plot test img
    vec_test.loc[test_img_idx]['filename']
    plotimg(pwd + '/data/test/' + vec_test.loc[test_img_idx]['filename'])

    # do dot prod
    test_img_vec = vec_test_num.loc[test_img_idx][:]
    test_img_vec = test_img_vec.reshape(len(test_img_vec),1)
    a = np.dot(vec_valid_num.ix[:], test_img_vec)

    # transform scores
    results = pd.DataFrame(a)
    results['filenames'] = vec_valid['filename']
    results.columns = ['scores', 'filenames']
    results.sort_values('scores', ascending = False, inplace = True)
    results.head()

    # get matches
    matches = results['filenames'].values[:10]
    match_scores = results['scores'].values[:10]

    plot_pic_grid(matches)
```

https://github.com/alexcnwy/CTDL_CNN_TALK_20170620

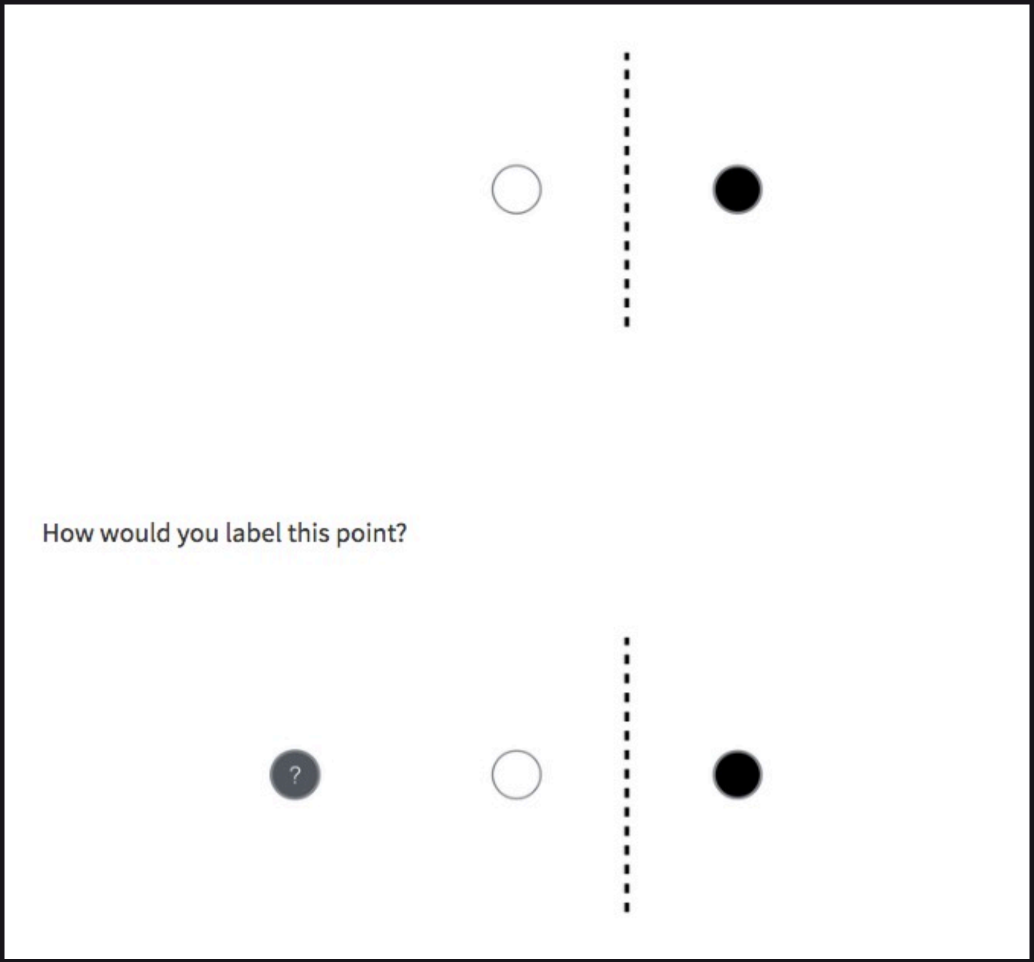


Practical Tips

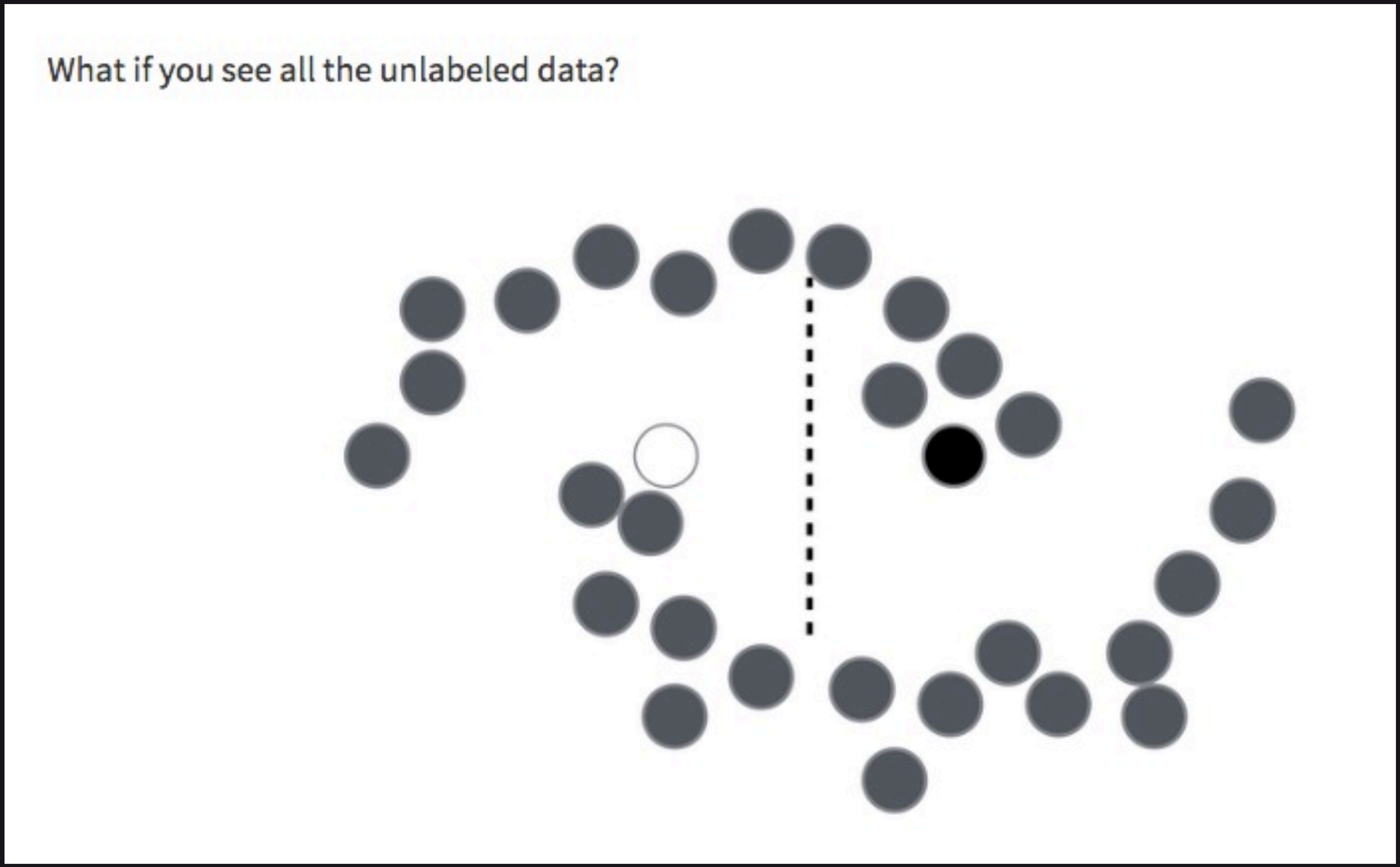
- use a GPU – AWS p2 instances not that expensive – much faster
- use “adam” / different optimizers SGD variants
  - http://sebastianruder.com/content/images/2016/09/saddle_point_evaluation_optimizers.gif
- look at nvidia-smi
- when overfitting - try raise dropout and stop training sooner
- when underfitting, try:

  1. Add more data
  2. Use data augmentation
     - flipping
     - slightly changing hues
     - stretching
     - shearing
     - rotation
  3. Use more complicated architecture (Resnets, Inception, etc)

Pseudo Labelling



Pseudo Labelling



Pseudo Labelling

Secure https://arxiv.org/abs/1503.02531

Cornell University Library

arXiv.org > stat > arXiv:1503.02531

Search or Art

(Help | Advanced

Statistics > Machine Learning

Distilling the Knowledge in a Neural Network

Geoffrey Hinton, Oriol Vinyals, Jeff Dean

(Submitted on 9 Mar 2015)

A very simple way to improve the performance of almost any machine learning algorithm is to train many different models on the same data and then to average their predictions. Unfortunately, making predictions using a whole ensemble of models is cumbersome and may be too computationally expensive to allow deployment to a large number of users, especially if the individual models are large neural nets. Caruana and his collaborators have shown that it is possible to compress the knowledge in an ensemble into a single model which is much easier to deploy and we develop this approach further using a different compression technique. We achieve some surprising results on MNIST and we show that we can significantly improve the acoustic model of a heavily used commercial system by distilling the knowledge in an ensemble of models into a single model. We also introduce a new type of ensemble composed of one or more full models and many specialist models which learn to distinguish fine-grained classes that the full models confuse. Unlike a mixture of experts, these specialist models can be trained rapidly and in parallel.

Comments: NIPS 2014 Deep Learning Workshop
Subjects: **Machine Learning (stat.ML)**; Learning (cs.LG); Neural and Evolutionary Computing (cs.NE)
Cite as: arXiv:1503.02531 **[stat.ML]**
(or arXiv:1503.02531v1 **[stat.ML]** for this version)

Image Cropping

Label the bounding boxes

Learn to predict them

Just extra input to CNN

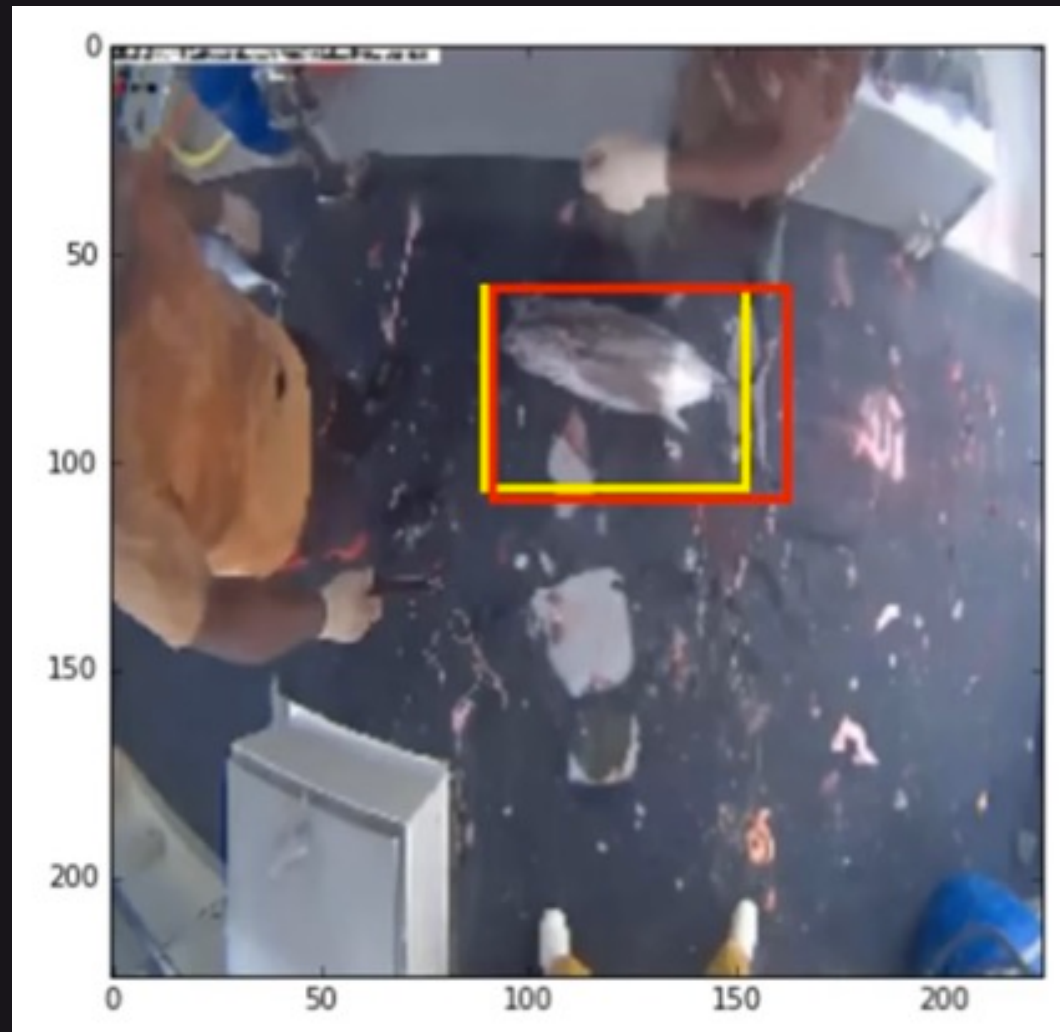
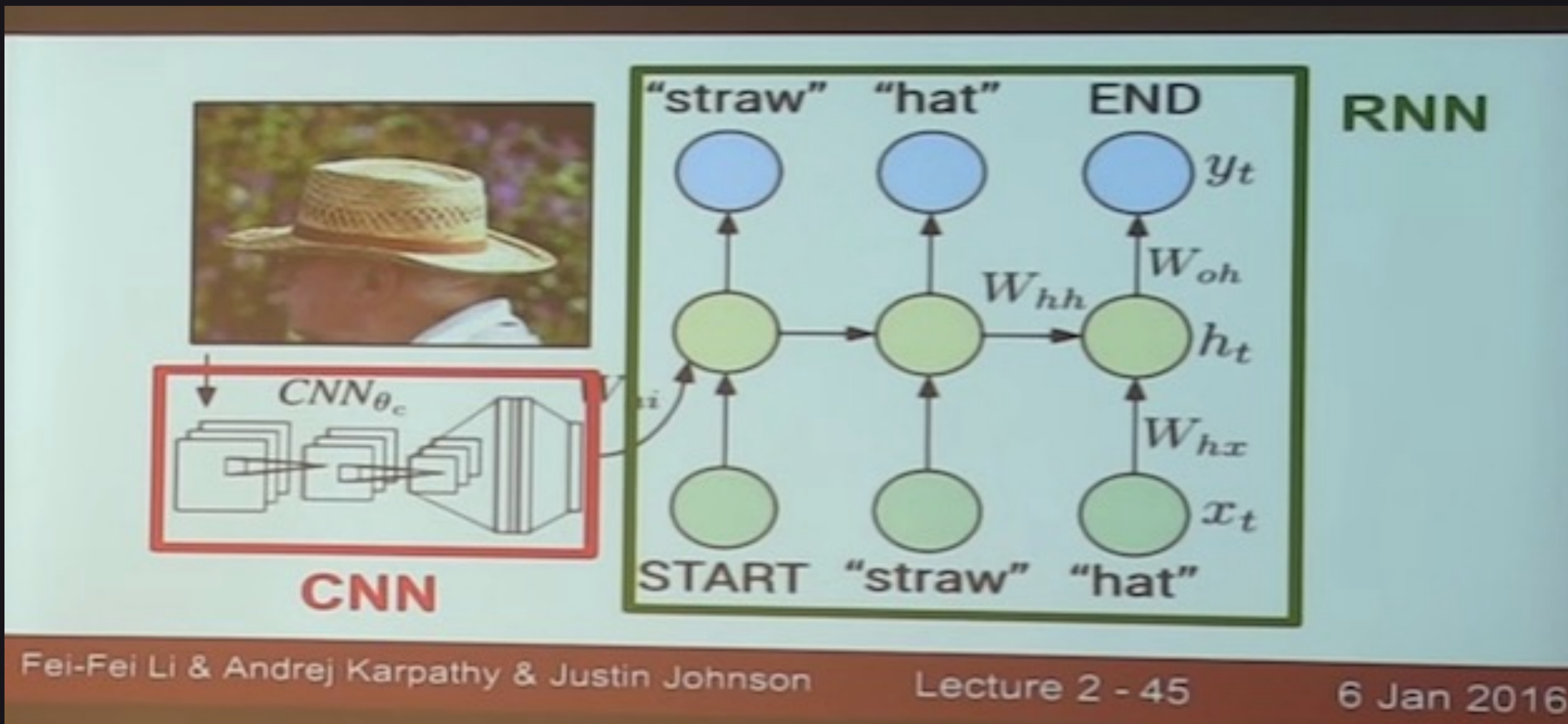
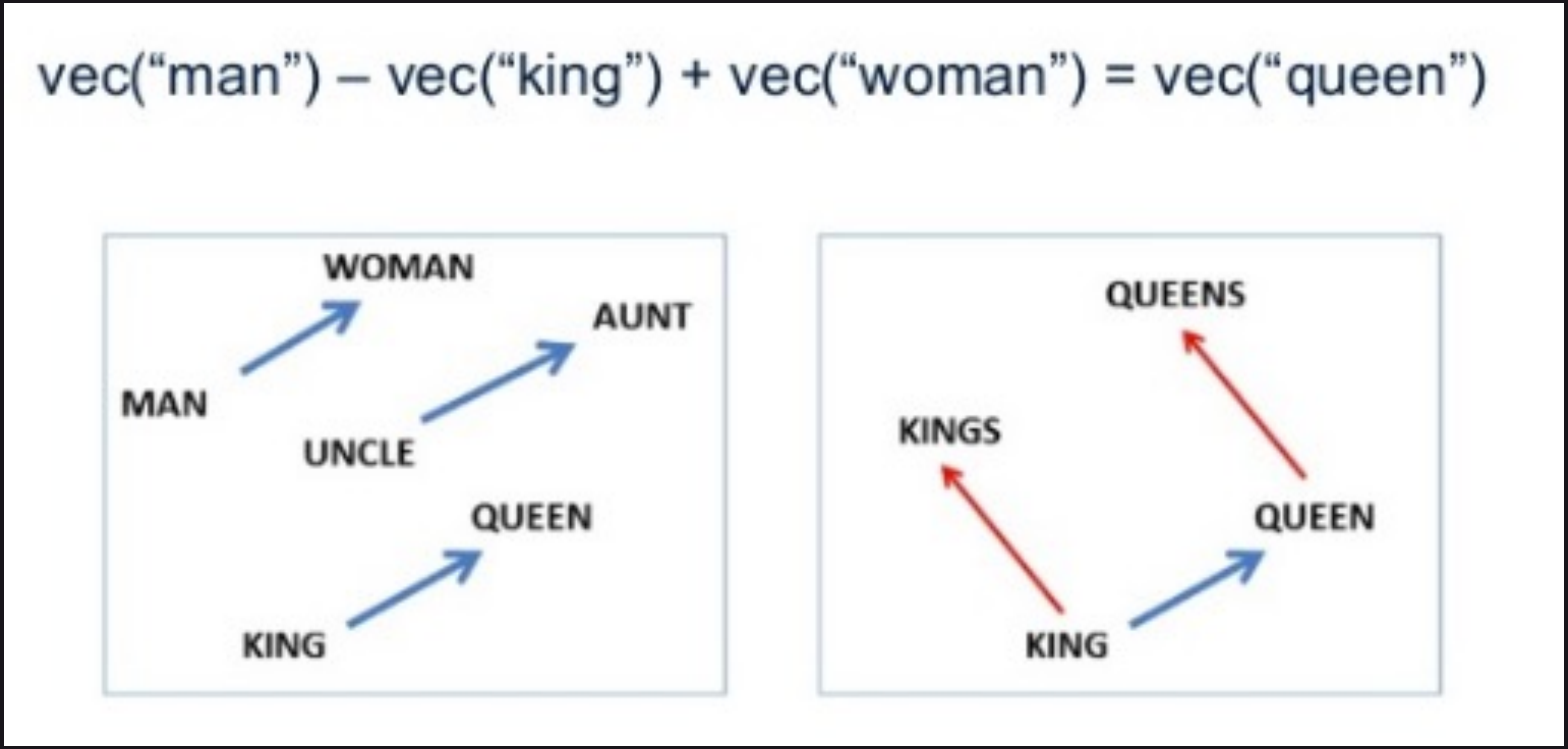
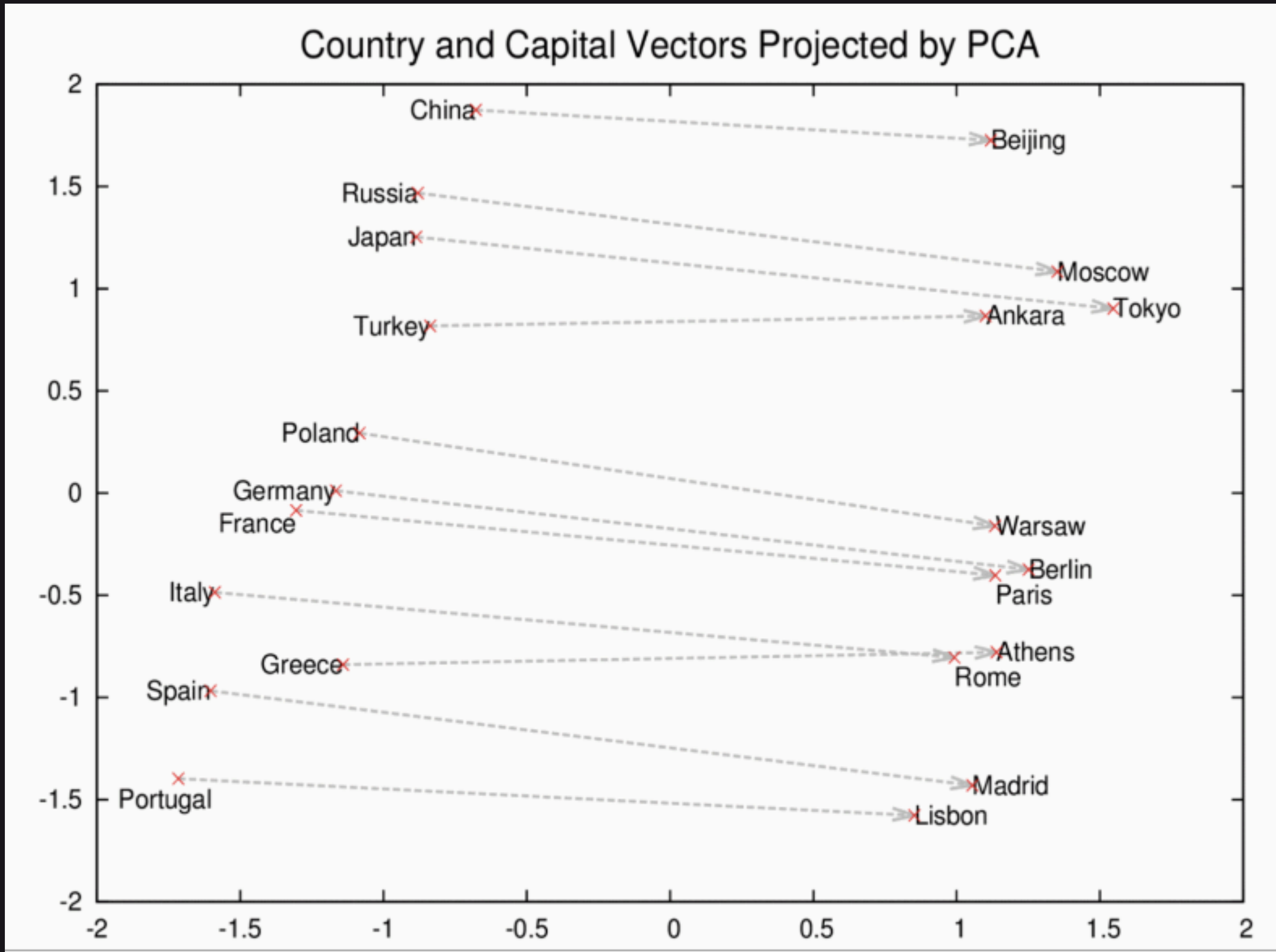


Image Captioning

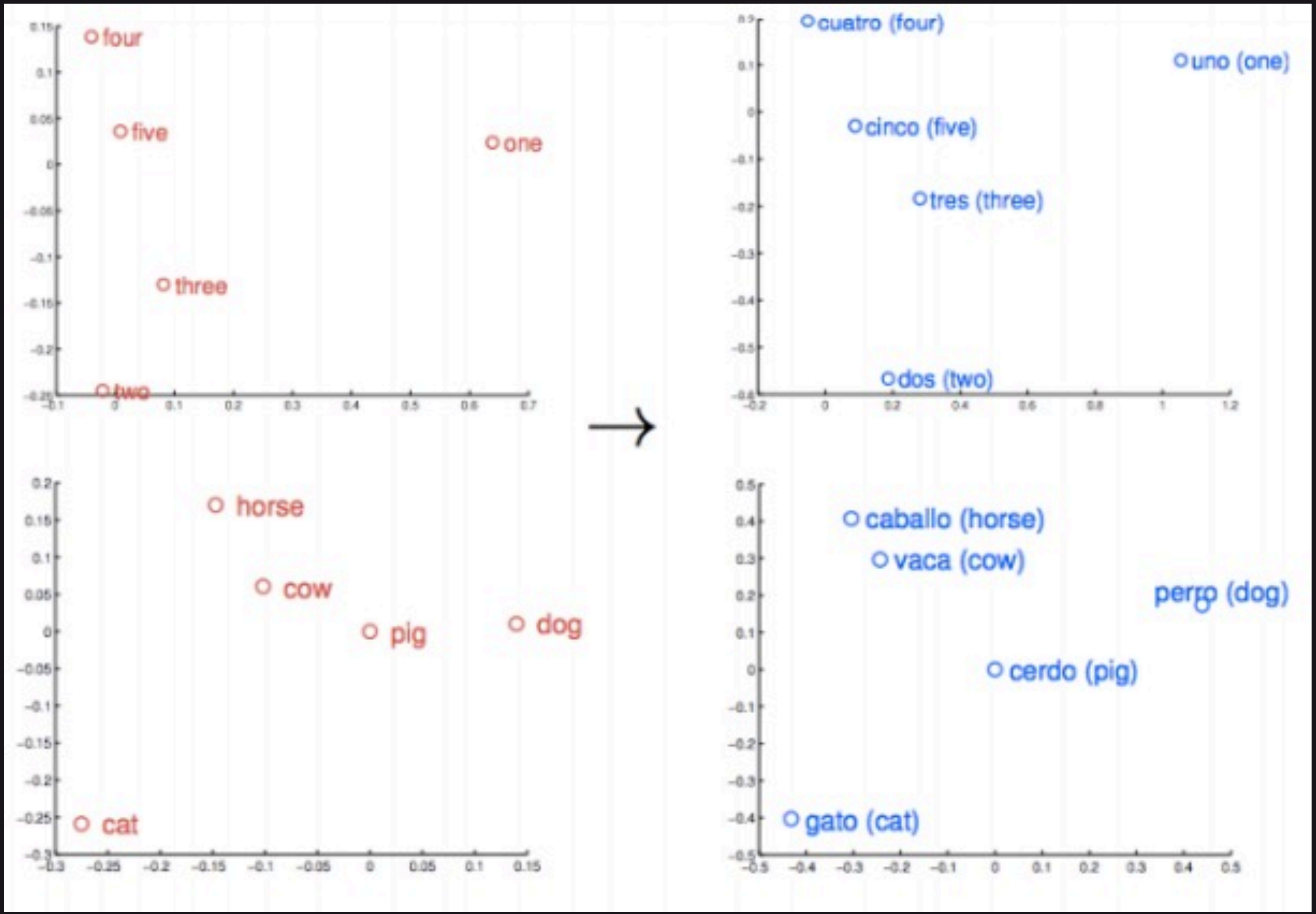


CNN + Word2Vec





Country and Capital Vectors Projected by PCA
China
Beijing
Russia
Japan
Moscow
Tokyo
Turkey
Ankara
Poland
Germany
France
Warsaw
Berlin
Paris
Italy
Greece
Spain
Athens
Rome
Portugal
Madrid
Lisbon
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2


four
five
one
three
two
cuatro (four)
uno (one)
cinco (five)
tres (three)
dos (two)
horse
cow
pig
dog
cat
caballo (horse)
vaca (cow)
perro (dog)
cerdo (pig)
gato (cat)

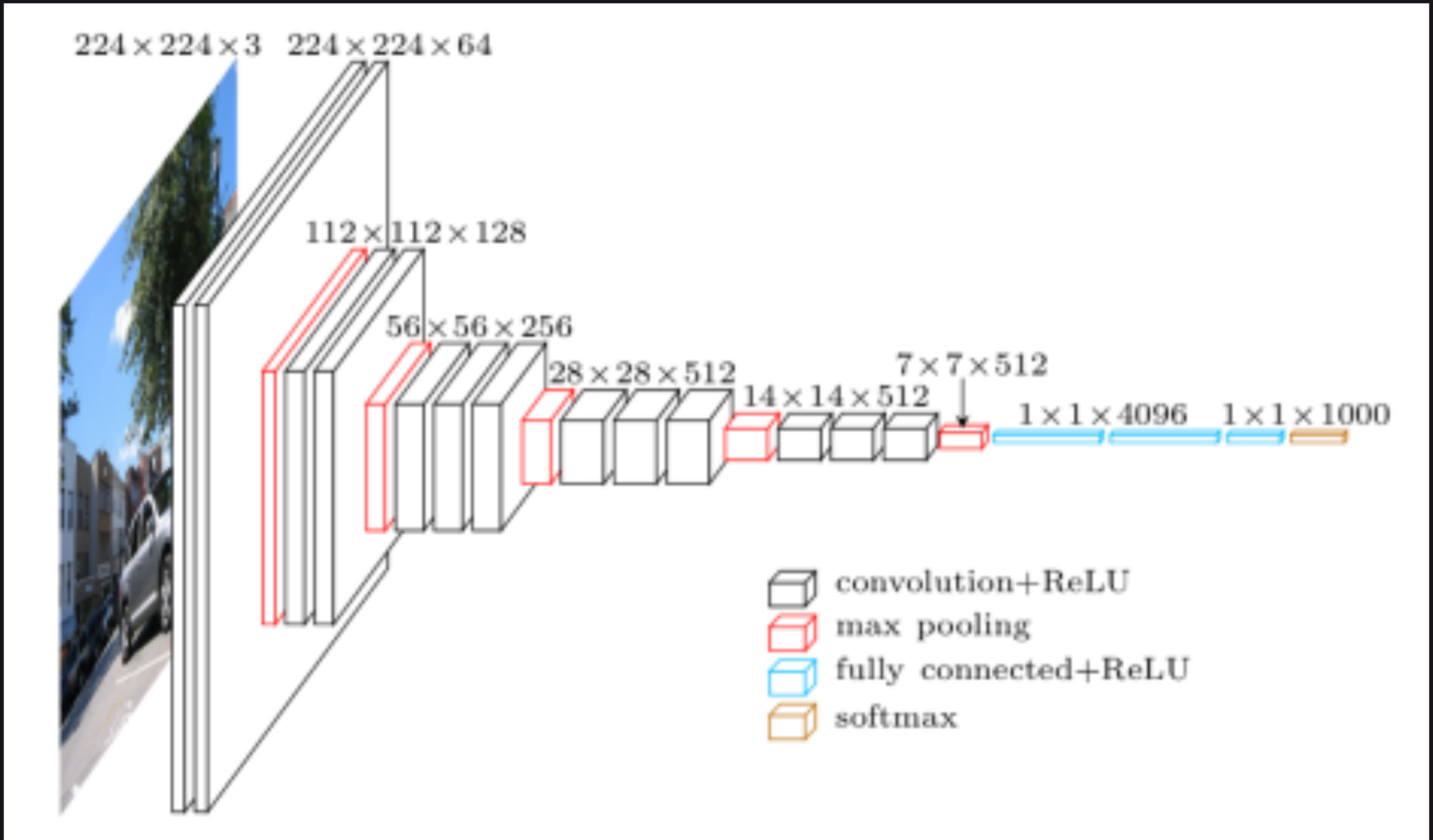
CNN + Word2Vec

DeViSE: A Deep Visual-Semantic Embedding Model

Andrea Frome*, Greg S. Corrado*, Jonathon Shlens*, Samy Bengio
Jeffrey Dean, Marc'Aurelio Ranzato, Tomas Mikolov
* These authors contributed equally.

{afrome, gcorrado, shlens, bengio, jeff, ranzato[†], tmikolov}@google.com
Google, Inc.
Mountain View, CA, USA

CNN + Word2Vec



- Learn the word2vec vectors for each ImangeNet noun

Style Transfer

- http://blog.romanofoti.com/style_transfer/

- https://github.com/junyanz/CycleGAN

Where to From Here?

- Clone the repo and train your own model
- Do the fast.ai course
- Read the cs231n notes
- Read http://colah.github.io/posts
- Email me questions /ideas :)
  alex@numberboost.com

THANKS!

https://github.com/alexcnwy/CTDL_CNN_TALK_20170620

Alex Conway
alex @ numberboost.com
NUMBERBOOST