Deep Learning: Day 3

chyld @ galvanize

Topics

———

- Install Python Modules
- Download Training Datasets
- Setup Kaggle
- Setup AWS EC2 for Deep Learning
- Image Classification with Real World data
- Transfer Learning
- Image Classification using Transfer Learning
- Multi-label Image Classification

Packages to Install

———

- `pip install -U scikit-learn pillow glances`
- `pip install -U opencv-python tqdm seaborn kaggle`

Kaggle: Dogs vs Cats

___

`https://www.kaggle.com/c/dogs-vs-cats`

`Get train.zip`

Dogs vs. Cats

Create an algorithm to distinguish dogs from cats

215 teams · 4 years ago

Overview   **Data**   Discussion   Leaderboard   Rules   Team

Competition Data

- sampleSubmission.csv
- test1.zip
- train.zip

train.zip 543.16 MB

⬇ Download

Kaggle: Understanding the Amazon from Space

— — —

https://www.kaggle.com/c/planet-understanding-the-amazon-from-space

Get train_v2.csv.zip & train-jpg.tar.7z

Kaggle

- https://www.kaggle.com

    - Register for account

- Get API key

    - https://www.kaggle.com/<username>/account

    - mv kaggle.json ~/.kaggle

- Install Kaggle CLI

    - pip install -U kaggle

Download Dogs vs Cats Images from Kaggle

———

- https://www.kaggle.com/c/dogs-vs-cats
- Accept rules
- https://www.kaggle.com/c/dogs-vs-cats/rules
- Download images using CLI/Terminal
- "kaggle competitions download -c dogs-vs-cats"

Amazon Web Services

— — —

Deep Learning AMI (Ubuntu) Version 10.0 - ami-e580c79d

Comes with latest binaries of deep learning frameworks pre-installed in separate virtual environments: MXNet, TensorFlow, Caffe, Caffe2, PyTorch, Keras, Chainer, Theano and CNTK. Fully-configured with NVidia CUDA, cuDNN and NCCL as well as Intel MKL-DNN

Root device type: ebs   Virtualization type: hvm   ENA Enabled: Yes

Select

64-bit

Model	NVIDIA Tesla V100 GPUs	GPU Memory	NVIDIA NVLink	vCPUs	Main Memory	Network Bandwidth	EBS Bandwidth
p3.2xlarge	1	16 GiB	n/a	8	61 GiB	Up to 10 Gbps	1.5 Gbps
p3.8xlarge	4	64 GiB	200 GBps	32	244 GiB	10 Gbps	7 Gbps
p3.16xlarge	8	128 GiB	300 GBps	64	488 GiB	25 Gbps	14 Gbps

NVIDIA Tesla V100 GPU

HP nVidia Tesla V100 16GB PCIE x16 876340-001 876908-001 Q2N68A

Be the first to review this product...   Ask Or Answer A Quest

In stock. Ships from United States. Most customers
Sold and Shipped by Compeve ⊙

- Manufacturer nVidia
- Model Tesla V100
- MPN 876340-001
- SKU 101000100503
- Warranty 30 days
- UPC 190017178622
- Content Video card, power cord
- Condition New, pulls

Sold and Shipped by: **Compeve**

$8,899.00

$20.00 Shipping (restrictions apply)

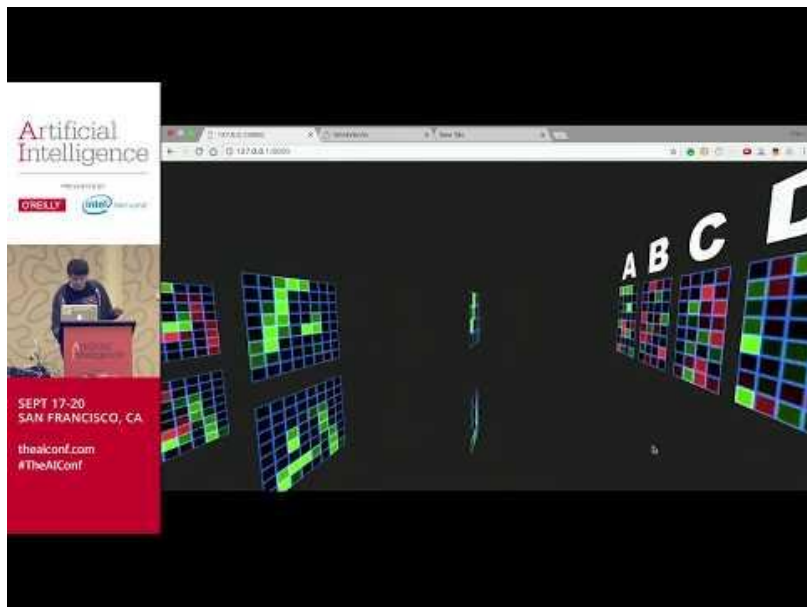| 1 | + |
| | − |

ADD TO CART ▶

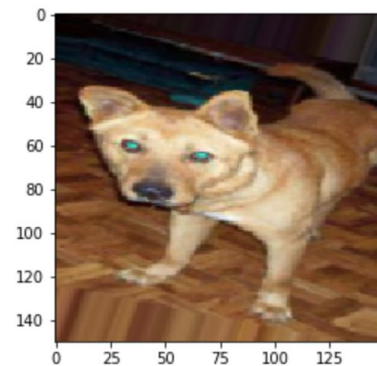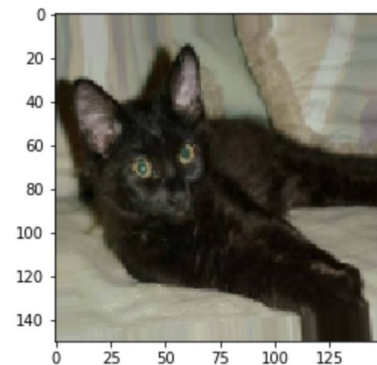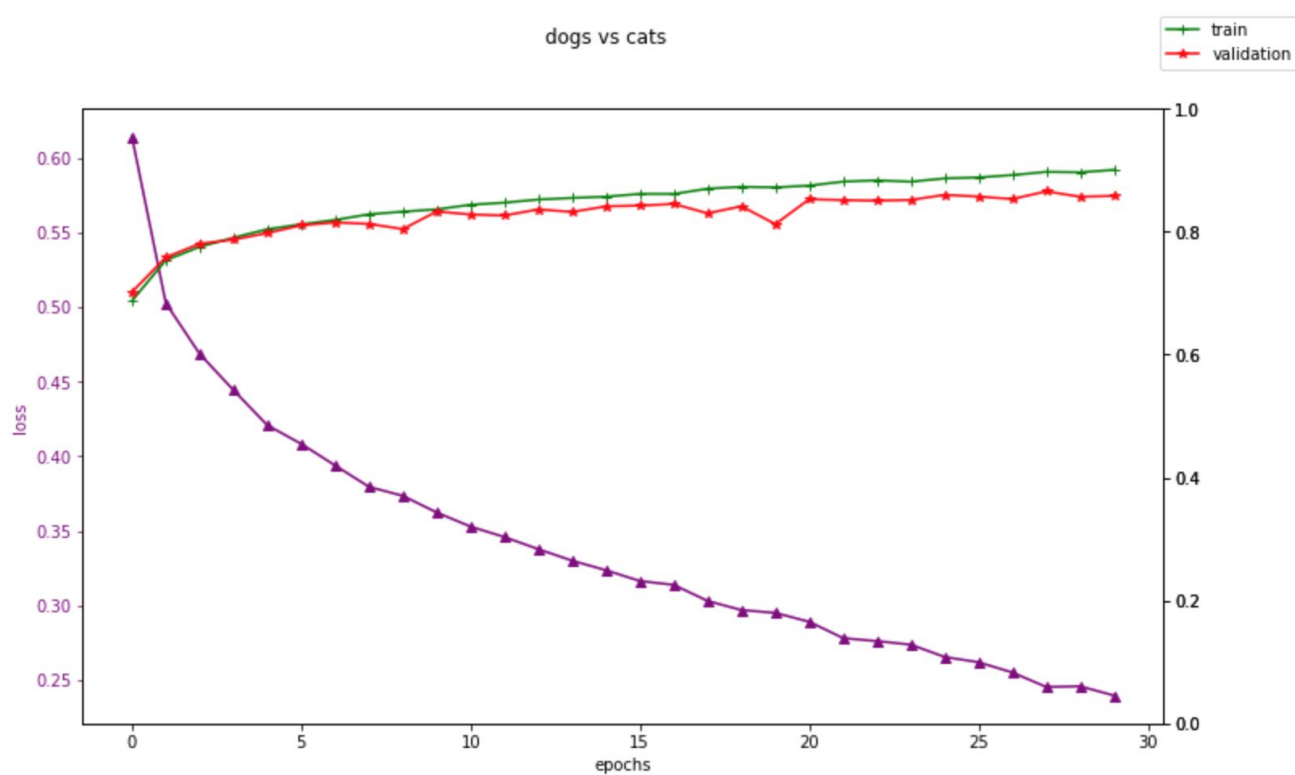☐ ADD TO COMPARE   🔔 PRICE ALERT

♥ ADD TO WISH LIST

Review: Visualizing ConvNet Architecture

———

- `https://youtu.be/0qm9vsf_hvU?t=265`

CNN: Real World Data

Transfer Learning

— — —

What is Transfer Learning?

Transfer learning, is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.
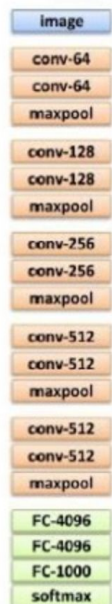
Transfer Learning

— — —

Why Transfer Learning?

- In practice a very few people train a Convolution network from scratch (random initialisation) because it is rare to get enough dataset. So, using pre-trained network weights as initialisations or a fixed feature extractor helps in solving most of the problems in hand.

- Very Deep Networks are expensive to train. The most complex models take weeks to train using hundreds of machines equipped with expensive GPUs.

- Determining the topology/flavour/training method/hyper parameters for deep learning is a black art with not much theory to guide you.

Transfer Learning

Transfer Learning with CNNs

1. Train on ImageNet

Network layers:
image → conv-64 → conv-64 → maxpool → conv-128 → conv-128 → maxpool → conv-256 → conv-256 → maxpool → conv-512 → conv-512 → maxpool → conv-512 → conv-512 → maxpool → FC-4096 → FC-4096 → FC-1000 → softmax
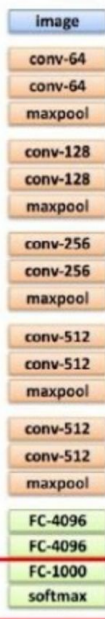
2. If small dataset: fix all weights (treat CNN as fixed feature extractor), retrain only the classifier

i.e. swap the Softmax layer at the end

[FC-1000, softmax highlighted]

3. If you have medium sized dataset, "finetune" instead: use the old weights as initialization, train the full network or only some of the higher layers
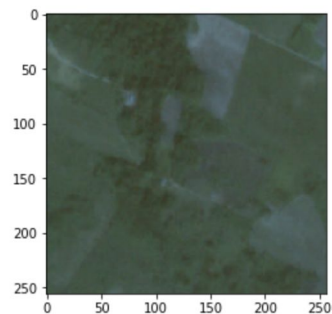
retrain bigger portion of the network, or even all of it.

[conv-512 ... softmax highlighted]

Transfer Learning
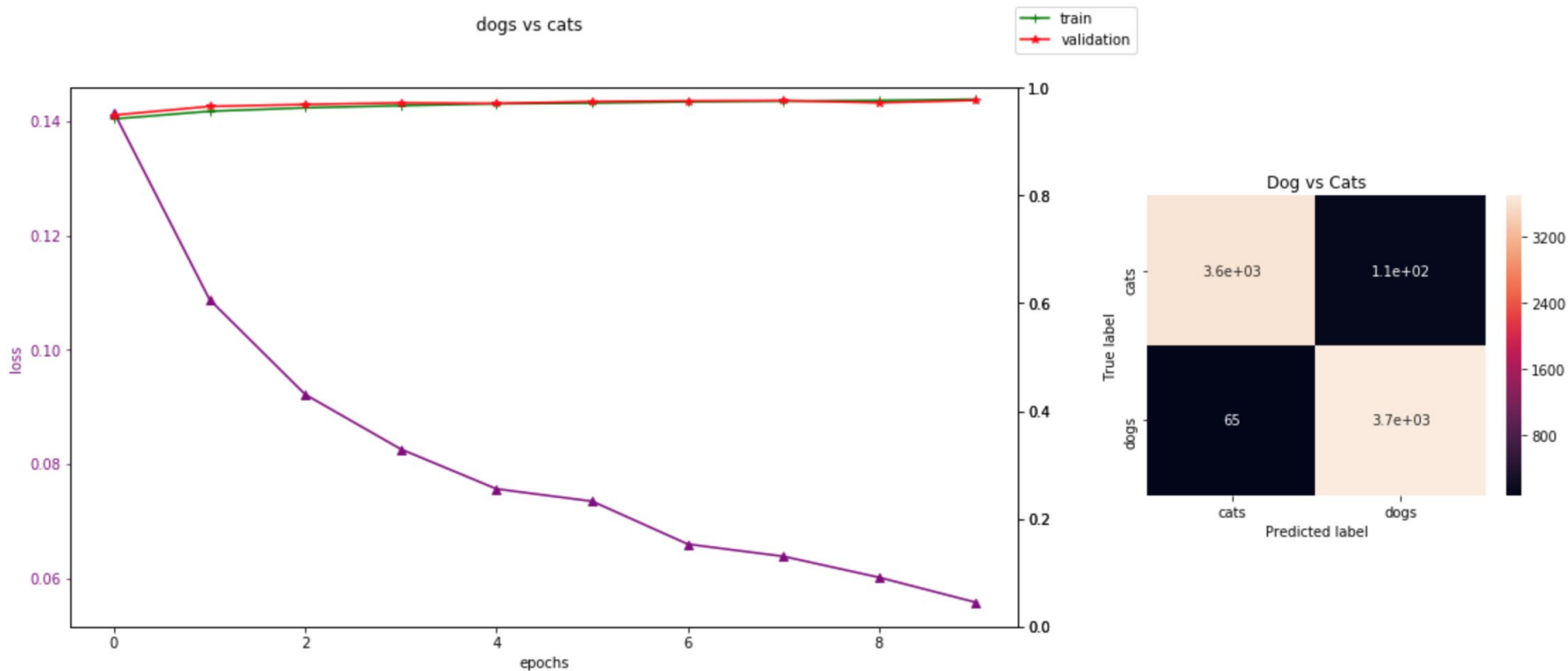
Multi-Label Classification

```
[(0.999877, 'clear'), (0.9303916, 'agriculture'), (0.9999151, 'primary'), (0.6120964, 'road')]
[(1, 'clear'), (1, 'agriculture'), (1, 'primary'), (1, 'road')]
```



```
[(1.0, 'clear'), (0.90402347, 'agriculture'), (1.0, 'primary'), (0.6447628, 'water')]
[(1, 'clear'), (1, 'agriculture'), (1, 'primary'), (1, 'cultivation'), (1, 'water')]
```