MURA Deep Learning Competition

A Case Study

ACSIP
IT 资深人协会
A Platform
For Senior IT Professionals
CHN IT Prof

ACSIP
IT 资深人协会

Why It Matters

- Is DL competition achievable by us, IT professionals?
- If so, how?
  - DL Knowledge
  - Data Science Methodologies
  - Software: Programing Language / DL Platform / Libraries / Tools
  - Hardware: GPU
  - Time commitment
- How to go beyond?

ACSIP 资深人协会

The MURA Paper

- https://arxiv.org/pdf/1712.06957.pdf

MURA: Large Dataset for Abnormality Detection in Musculoskeletal Radiographs

Pranav Rajpurkar[1,*], Jeremy Irvin[1,*], Aarti Bagul[1], Daisy Ding[1], Tony Duan[1], Hershel Mehta[1], Brandon Yang[1], Kaylie Zhu[1], Dillon Laird[1], Robyn L. Ball[2], Curtis Langlotz[3], Katie Shpanskaya[3], Matthew P. Lungren[3,†], Andrew Y. Ng[1,†]

*,†Equal Contribution

[1] Department of Computer Science
Stanford University
{pranavsr, jirvin16}@cs.stanford.edu

[2]Department of Medicine
Stanford University
rball@stanford.edu

[3]Department of Radiology
Stanford University
mlungren@stanford.edu

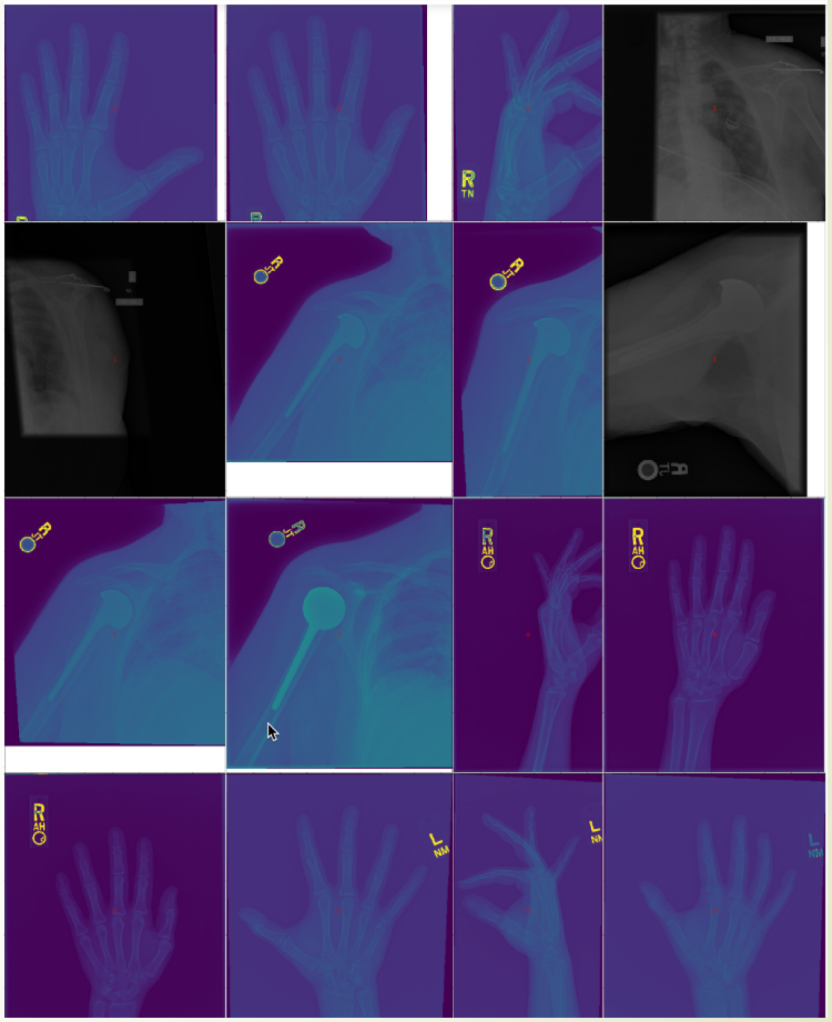
Abstract

We introduce MURA, a large dataset of musculoskeletal radiographs containing 40,561 images from 14,863 studies, where each study is manually labeled by radiologists as either normal or abnormal. To evaluate models robustly and to get an estimate of radiologist performance, we collect additional labels from six board-certified Stanford radiologists on the test set, consisting of 207 musculoskeletal studies. On this test set, the majority vote of a group of three radiologists serves as gold standard. We train a 169-layer DenseNet baseline model to detect and localize abnormalities. Our model achieves an AUROC of 0.929, with an operating point of 0.815 sensitivity and 0.887 specificity. We compare our model and radiologists on the Cohen's kappa statistic, which expresses the agreement of our model and of each radiologist with the gold standard. Model performance is comparable to the best radiologist performance in detecting abnormalities on finger and wrist studies. However, model performance is lower than best radiologist performance in detecting abnormalities on elbow, forearm, hand, humerus, and shoulder studies. We believe that the task is a good challenge for future research. To encourage advances, we have made our dataset freely available at http://stanfordmlgroup.github.io/competitions/mura.

ACSIP
IT 资深人协会

MURA – What Is It?

- Musculoskeletal Radiographs
- 40k bone X-Rays
- Seven Body Parts
  - Elbow
  - Finger
  - Hand
  - Forearm
  - Humerus
  - Shoulder
  - Wrist



ACSIP
IT 资深人协会

Sample X-Rays









ACSIP 资深人协会

The Baseline

	Radiologist 1	Radiologist 2	Radiologist 3	Model
Elbow	0.850 (0.830, 0.871)	0.710 (0.674, 0.745)	0.719 (0.685, 0.752)	0.710 (0.674, 0.745)
Finger	0.304 (0.249, 0.358)	0.403 (0.339, 0.467)	0.410 (0.358, 0.463)	0.389 (0.332, 0.446)
Forearm	0.796 (0.772, 0.821)	0.802 (0.779, 0.825)	0.798 (0.774, 0.822)	0.737 (0.707, 0.766)
Hand	0.661 (0.623, 0.698)	0.927 (0.917, 0.937)	0.789 (0.762, 0.815)	0.851 (0.830, 0.871)
Humerus	0.867 (0.850, 0.883)	0.733 (0.703, 0.764)	0.933 (0.925, 0.942)	0.600 (0.558, 0.642)
Shoulder	0.864 (0.847, 0.881)	0.791 (0.765, 0.816)	0.864 (0.847, 0.881)	0.729 (0.697, 0.760)
Wrist	0.791 (0.766, 0.817)	0.931 (0.922, 0.940)	0.931 (0.922, 0.940)	0.931 (0.922, 0.940)
Overall	0.731 (0.726, 0.735)	0.763 (0.759, 0.767)	0.778 (0.774, 0.782)	0.705 (0.700, 0.710)

ACSIP
IT 资深人协会

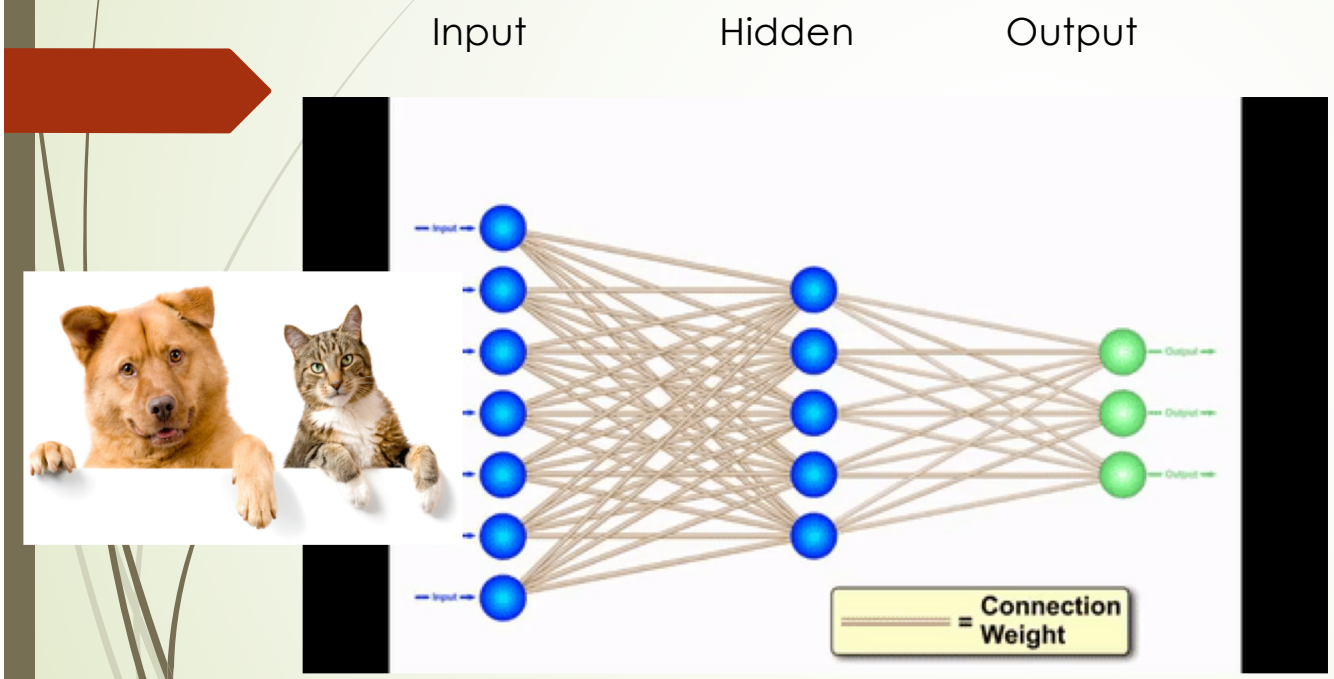
MURA Competition

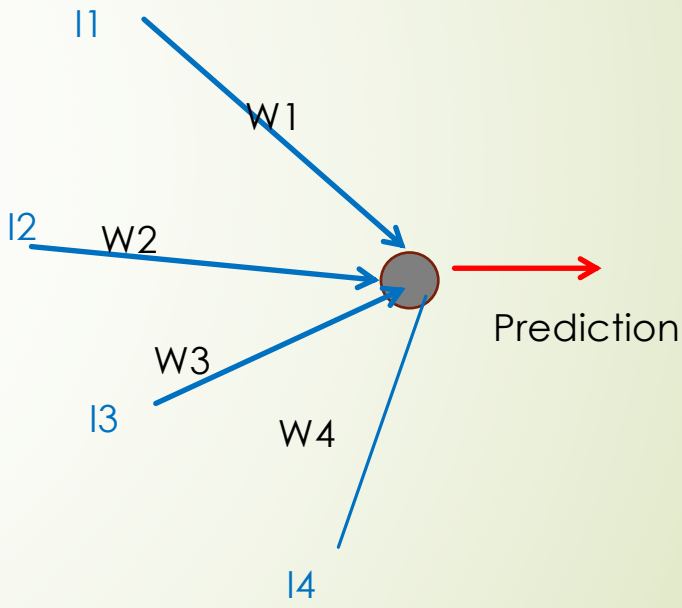
- A Good Way To Study
  - Baseline reference
  - Existing Paper
  - Simple Binary Classifications

HOW DO WE START FROM HERE?

ACSIP
IT 资深人协会

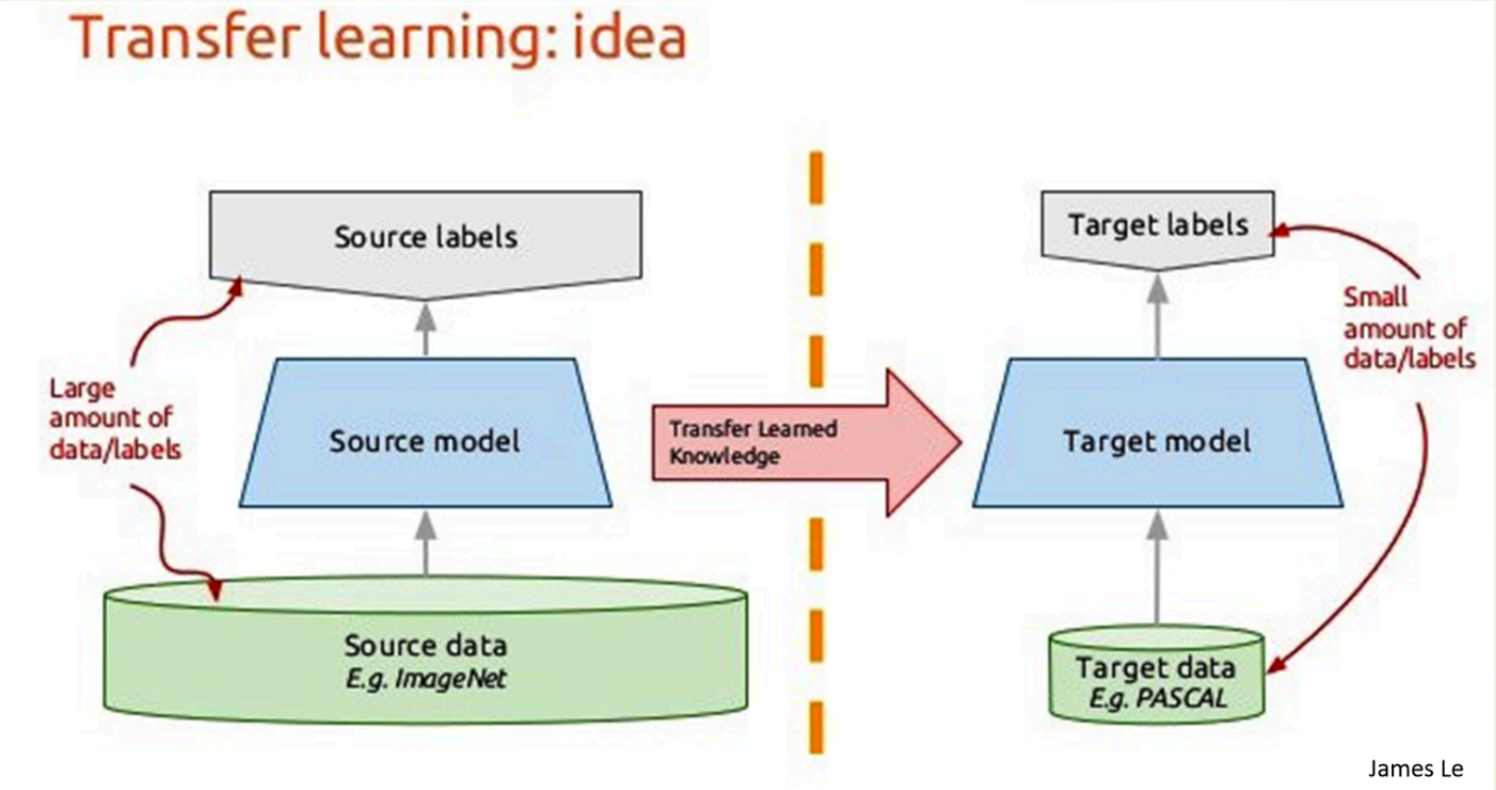
Neural Network Basis





ACSIP
IT 资深人协会

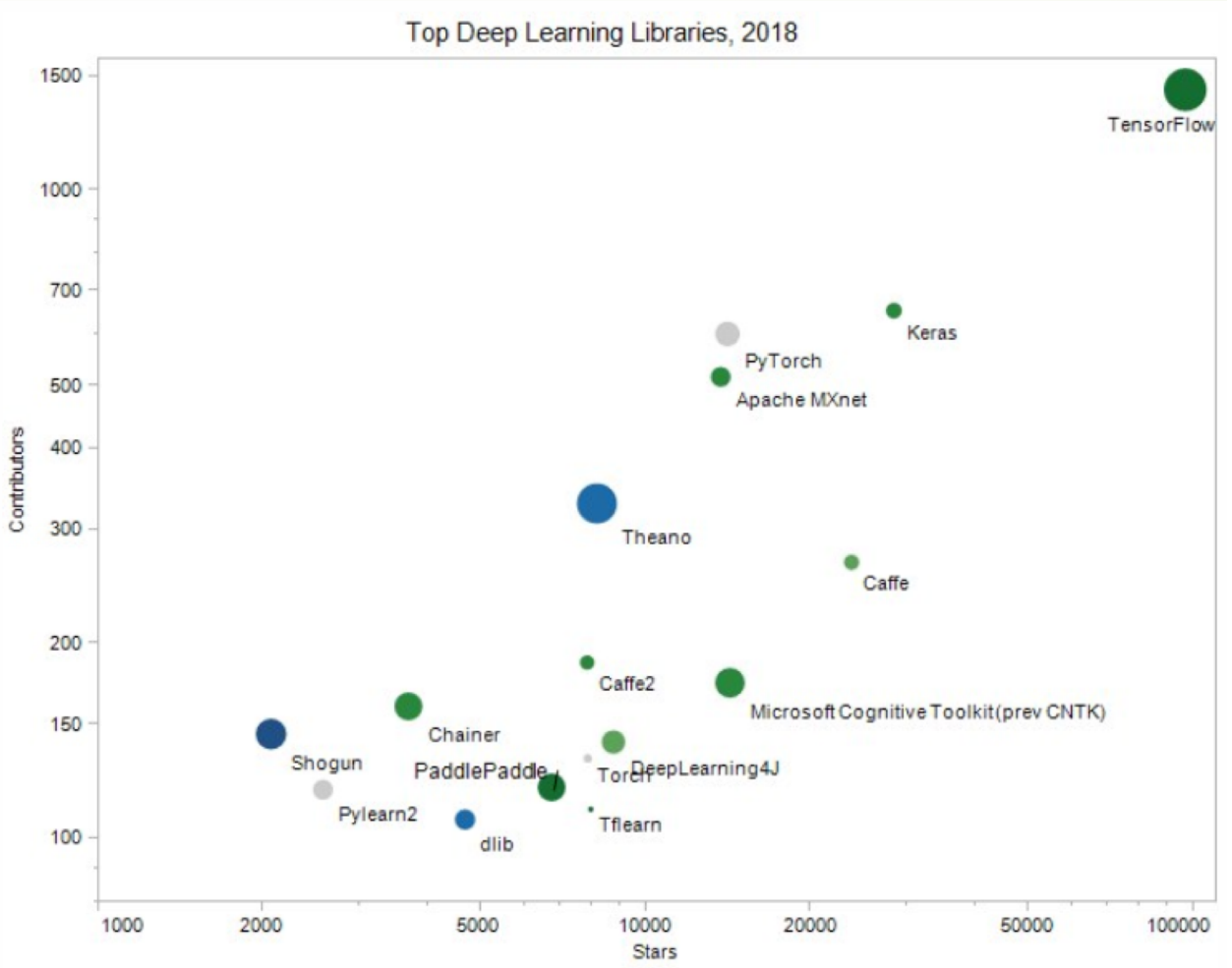
Transfer Learning



ACSIP 资深人协会

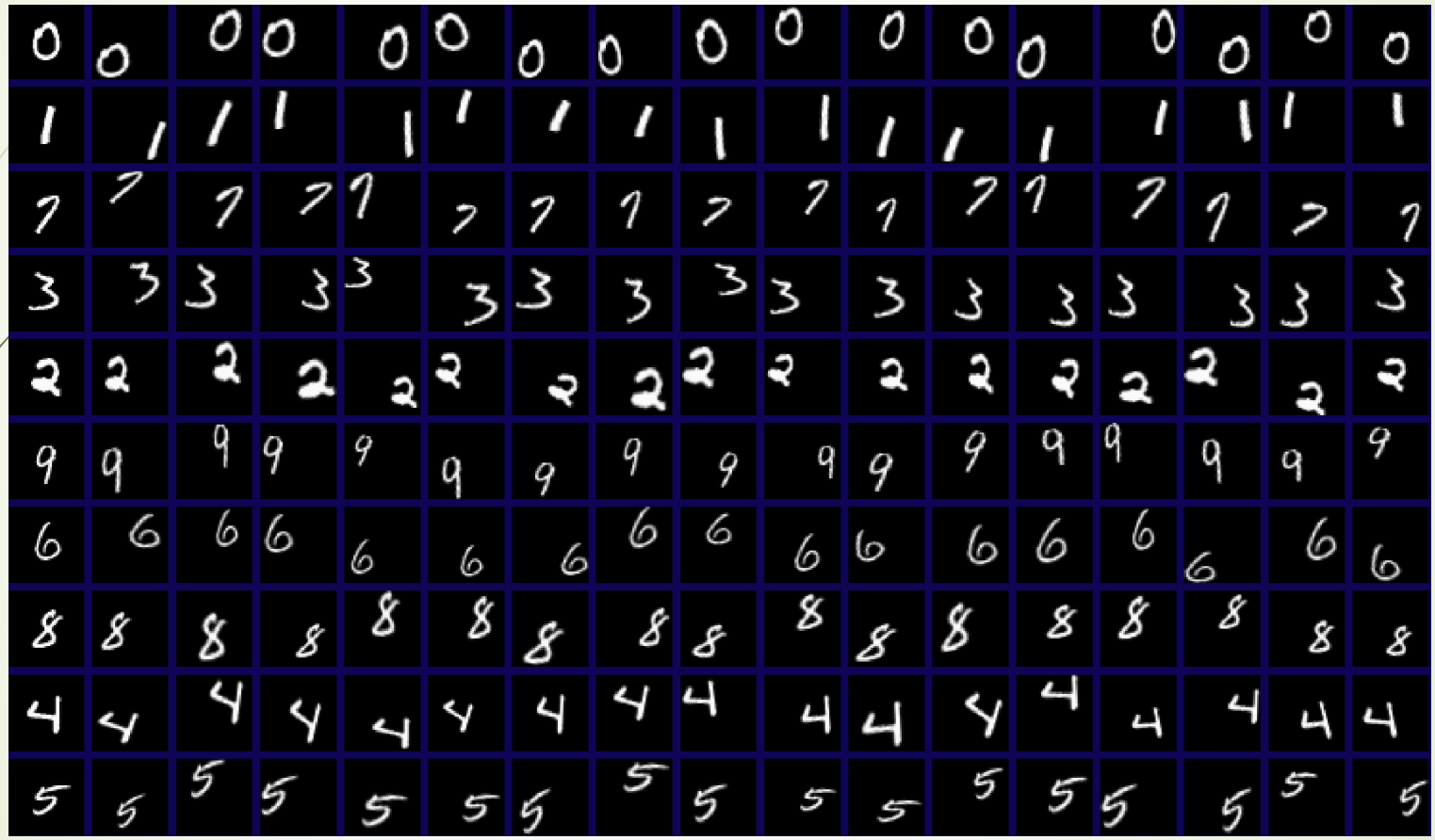
Platforms

- Tensorflow - Google
- Pytorch - Facebook
- MxNet - Apache
- CNTK – Microsoft
- Keras - François Chollet
- PaddlePaddle - Baidu



ACSIP
IT 资深人协会

MNIST



ACSIP
IT 资深人协会

MNIST – Load Data

Load MNIST data

```python
from keras.datasets import mnist

# Load pre-shuffled MNIST data into train and test sets
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

```python
print X_train.shape
# (60000, 28, 28)
```

ACSIP
IT 资深人协会

MNIST – Data Pre-Processing

Reshape input data — Python

```python
X_train = X_train.reshape(X_train.shape[0], 1, 28, 28)
X_test = X_test.reshape(X_test.shape[0], 1, 28, 28)
```

Convert data type and normalize values — Python

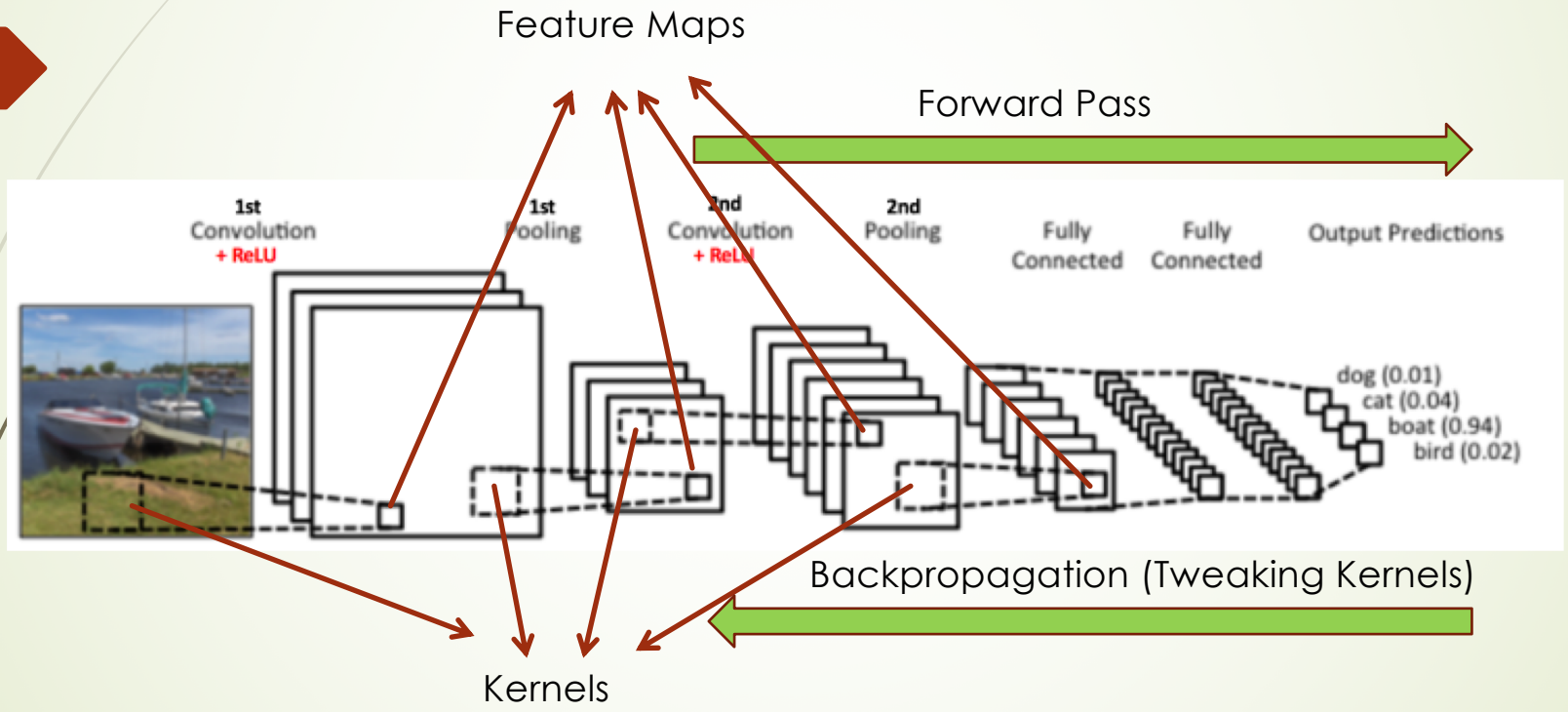
```python
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train /= 255
X_test /= 255
```

Preprocess class labels — Python

```python
# Convert 1-dimensional class arrays to 10-dimensional class matrices
Y_train = np_utils.to_categorical(y_train, 10)
Y_test = np_utils.to_categorical(y_test, 10)
```

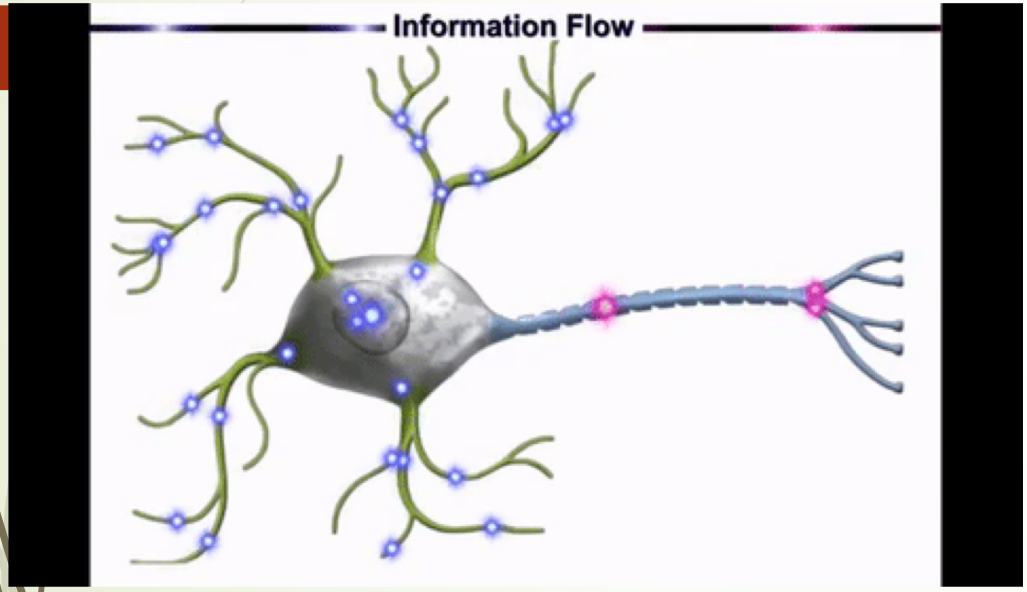
ACSIP
IT 资深人协会

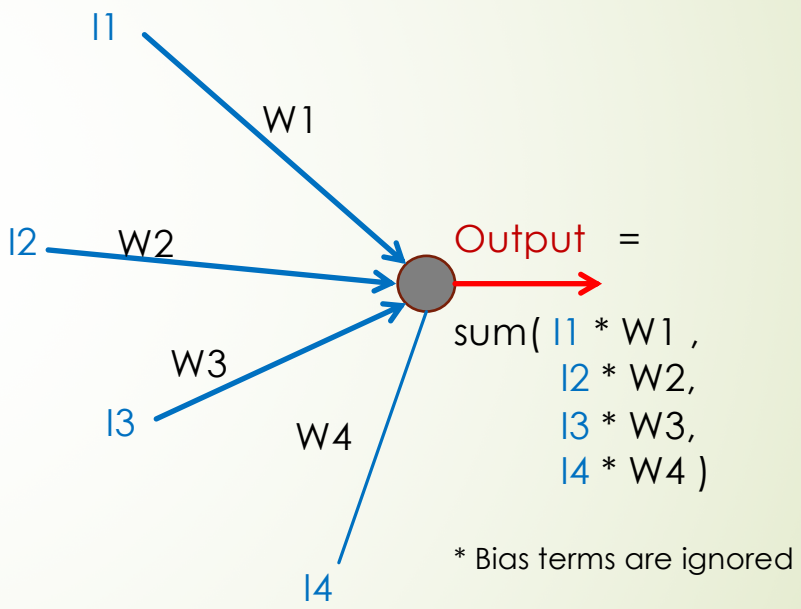
Convolutional Network Architecture (CNN)



ACSIP
IT 资深人协会

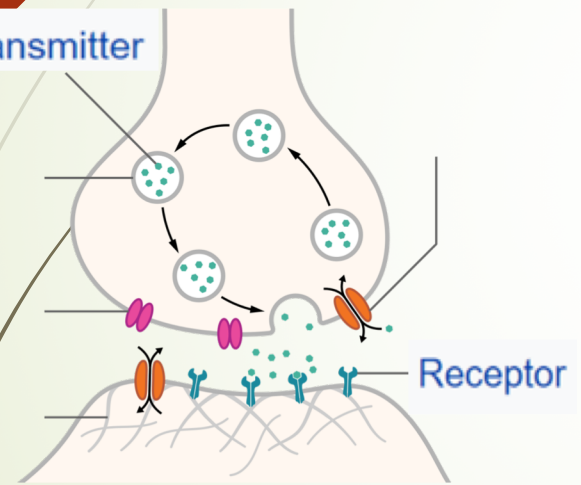
Human Neuron vs. Artificial Neuron

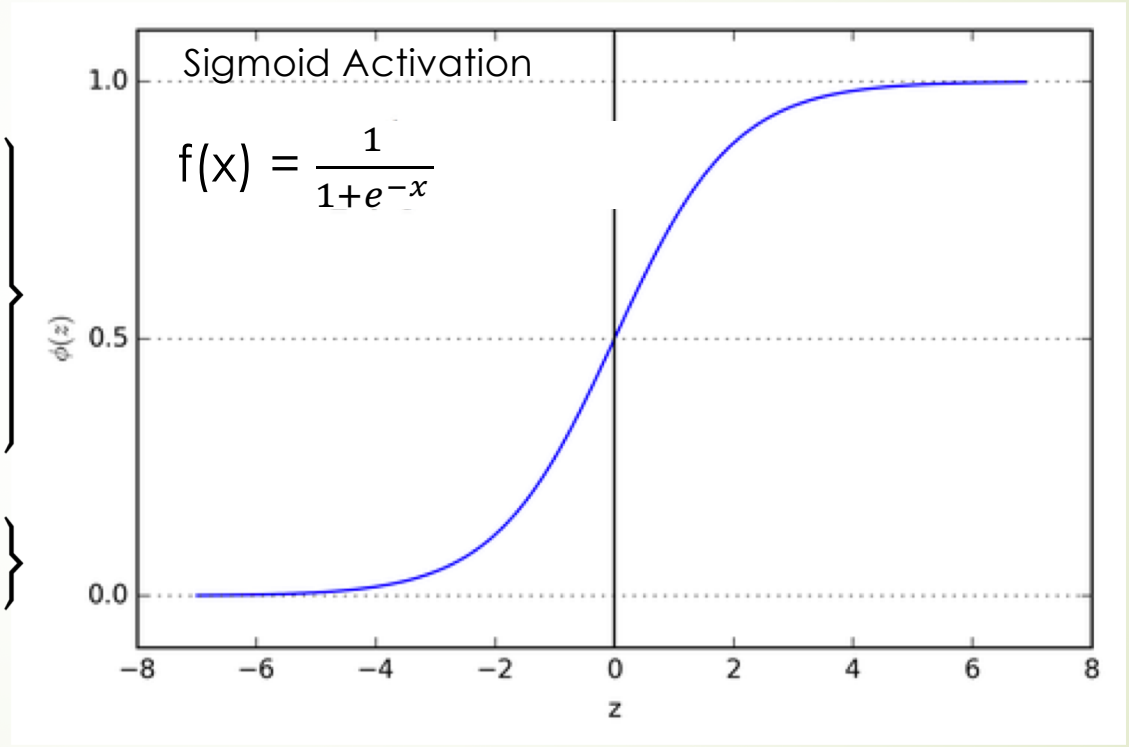




ACSIP
IT 资深人协会

Activation Function - Synapse & Receptor





ACSIP 资深人协会

MNIST - Model

Model architecture — Python

```python
model = Sequential()

model.add(Convolution2D(32, 3, 3, activation='relu', input_shape=(1,28,28)))
model.add(Convolution2D(32, 3, 3, activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax'))
```

ACSIP
IT 资深人协会

MNIST – Compile and Train

Compile model — Python

```
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

Fit Keras model — Python

```
model.fit(X_train, Y_train,
          batch_size=32, nb_epoch=10, verbose=1)
# Epoch 1/10
# 7744/60000 [==>..........................] - ETA: 96s - loss: 0.5806 - acc: 0.8164
```

ACSIP
IT 资深人协会

MNNIST - Evaluation

Evaluate Keras model — Python

```python
score = model.evaluate(X_test, Y_test, verbose=0)
```

ACSIP
IT 资深人协会

GPU Battle

- Build Your Own vs Cloud

Build Your Own:

- Full control
- Fixed cost (as opposed to Cloud)
- OS set up is preserved over time
- Cheaper than cloud over the long term
- Looks good

Cloud:

- Much more scalable
- Don't need to pay for electricity consumed by GPUs
- Don't need to worry about hardware issues
- Easy and secure remote access

Standard practice:

Set up model training pipeline and do experiments on local GPU, then spin up hundreds of cloud instances to perform hyperparameter tuning in parallel

ACSIP
IT 资深人协会

GPU - Cloud

	AWS	Paperspace	FloydHub	Suggested platform
Ease of setup	Complex. Manual setup required	Pre-installed libraries and frameworks	Provides almost all major DL environments	FloydHub/Paperspace
User Experience	Smooth. Data upload/download tricky	Quite okay apart from the server lag	Not very intuitive. Takes time getting used to	AWS
Hardware/Software	Tesla K80s	Latest Pascal and Volta series GPUs	Tesla K80s	Paperspace
Performance	Decent	Maxwell series – almost like AWS; Pascal series – 3x of AWS	0.75x of AWS	AWS/Paperspace
Additional features	Offers multi-GPU systems	Desktop environment; Shareable drives	Public datasets; concurrent job runs; support for MOOCs	FloydHub
Pricing	Starting at $0.9/hr + $13/mo for storage + IP	Starting at $0.4/hr + $7/mo for storage + IP	$14/mo + Powerups on base plan	Paperspace

ACSIP
IT 资深人协会

The Winner – Google Cloud

- US$300 Credit = 500 + hours Free GPU time if pick Tesla K80 (0.45 per hour)
- Excellent customer service
- Easy to setup and scalable
- Simple Pricing Model
  - Using it 25% of a month, standard discount kicks in
  - If running for full month, 30% discount

ACSIP
IT 资深人协会

Software Installation In the Cloud

- Ubuntu 16.04 LTS
- Python 3.5.2 or newer
- Tensorflow 1.8.0 or newer
- Keras 2.1.5 or newer
- Jupyter Notebook
- Numpy, Scipy, Scikit Learn, Scikit Image, Pandas, Matplotlib, Pillow
- If you prefer to use docker, there are plenty of docker image files to choose from.
- Keras docker https://github.com/keras-team/keras/tree/master/docker
- All in one DL docker https://github.com/floydhub/dl-docker

ACSIP
IT 资深人协会

Things to consider if you want to build your own workstation

- Nvidia vs AMD

NVIDIA:

- Much larger DL community
- Easy to find resources for
- Supports DL with 9 series + cards

AMD:

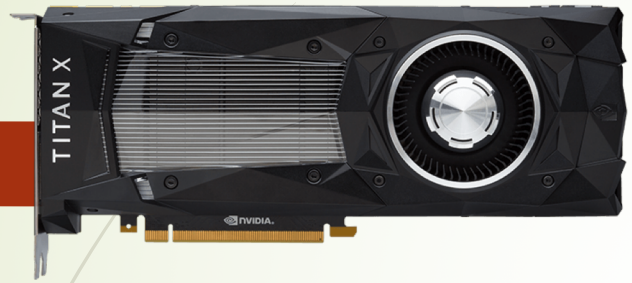
- More cost efficient
- Started DL cards rather recently so smaller community and less support
- Supports DL with Vega and later cards

Why not Intel?

Intel compute cards (Xeon Phi) are just way too expensive…

ACSIP
IT 资深人协会

GPU – Build Your Own



TITAN X
$1,600



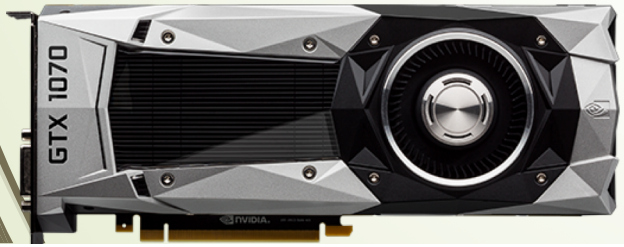
TITAN V
$3,000 - $5,000



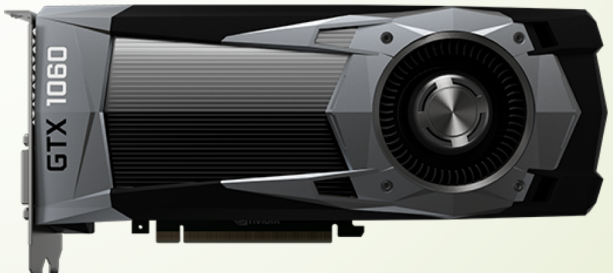
RTX 2080TI
$1,600



GTX 1080TI
$1,000



GTX 1070
$600



GTX 1060 (6GB)
$350

ACSIP
IT 资深人协会

An example of an overkill build
(And this is why you should define your budget early)



- Motherboard:
  - ASUS ROG Rampage V Edition 10
- CPU:
  - Intel i7 – 6850K
- RAM:
  - Corsair Dominator Platinum 4 x 16GB 2400GHz
- GPU:
  - 2 x EVGA Titan X Hybrid (Maxwell)
  - 2 x EVGA 1080 Ti FTW3 Hybrid
- Hard Drive:
  - Samsung 850 Evo M.2. SSD
  - 3 x Western Digital 4TB hard drive (RAID 5)

ACSIP
IT 资深人协会

GPU - BRANDS

- Brand
  - You may see these GPUs having a lot of different brands, such as
    - EVGA
    - ASUS
    - MSI
    - GIGABYTE
    - Etc.
  - Essentially NVIDIA/AMD only provides the chip, and each brand deals with
    - Overclocking
    - Cooling
    - Card Structure
  - Hence, check the price and review across different brands to decide which one to buy

GPU – HOW TO PICK?

GENERAL RULES:

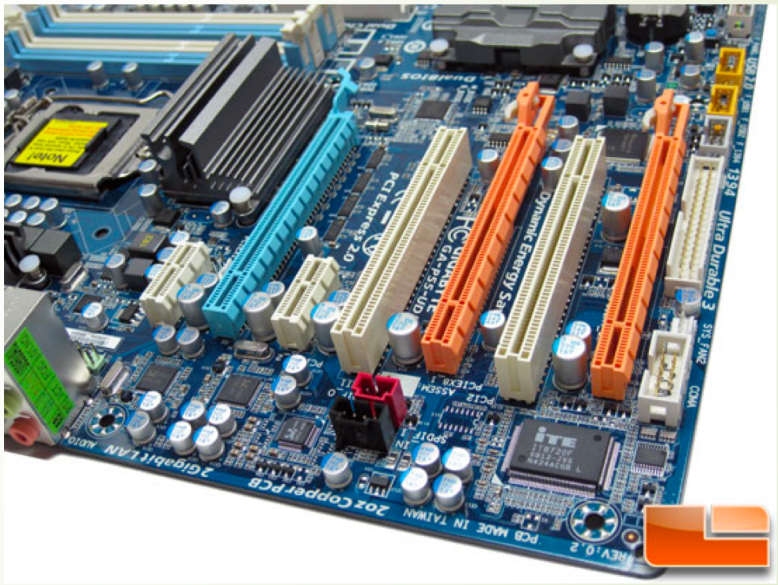
- With a budget, pick the one with best single precision TFLOPS spec
- Or, if GRAM matters:
  - Titan V (32GB) / Titan X (12GB) / RTX 2080TI (11GB) / GTX 1080TI (11GB)

- TFLOPS (Tera Floating-point Operations Per Second):
  - A measure of GPU compute performance
  - For entry-level Deep Learning, compare GPUs by their single-precision TFLOPS number, as that's the usual precision we use for the model
    - Double-precision is an overkill, and the fitting infrastructure for half-precision isn't common yet
  - However, if you would like to explore speeding up your training by using half-precision, check out the GPUs with TPUs (Tensor Processing Units)

ACSIP
IT 资深人协会

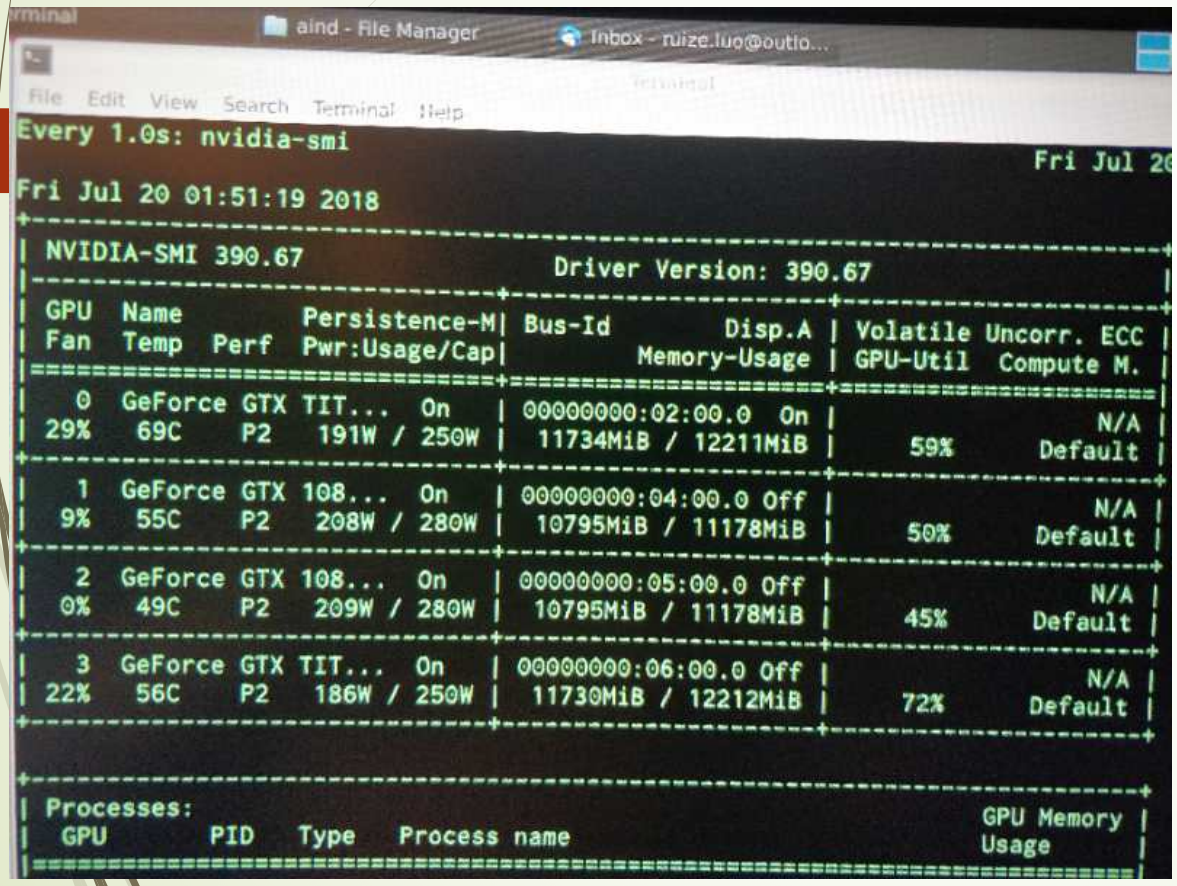
GPU - Multiple

- Multiple GPUs
  - For starters, consider ONE
  - The Lowest GPU is your bottleneck
  - Cooling System is Important!
    - 4 x GTX 1080 Ti = 1120W, a heater 24/7
  - Motherboard is important
    - Physical
    - Bandwidth
  - CPU (40 vs 28 PCIe Lanes)
  - PSU (1300W for 4 GPUs)
  - Case



ACSIP 资深人协会

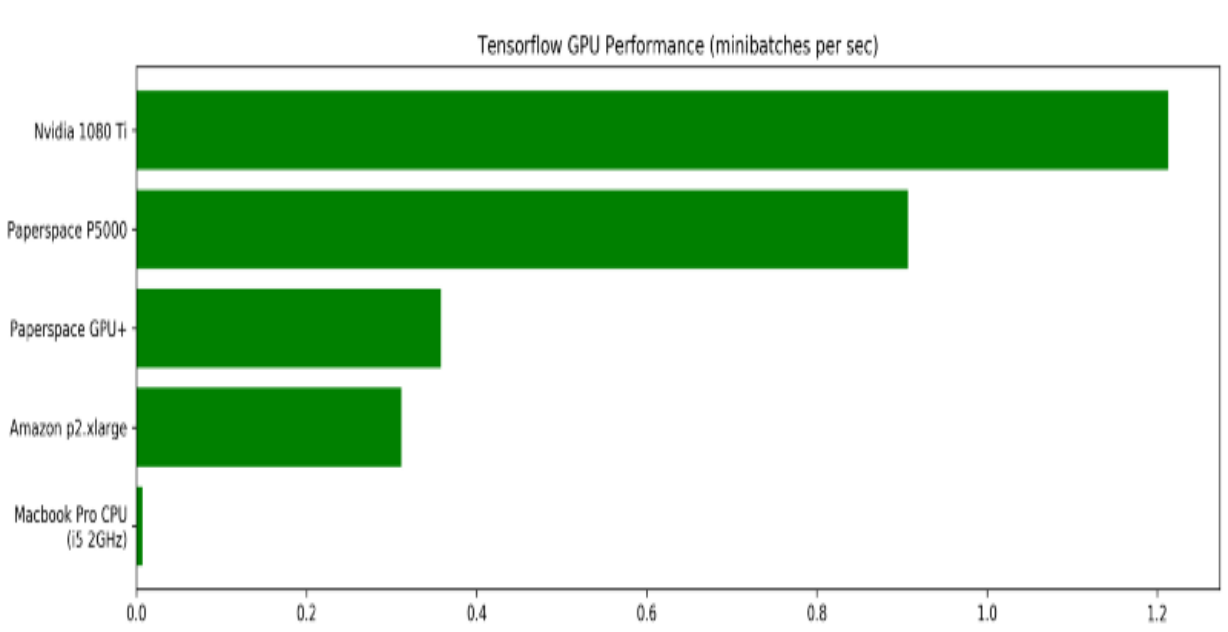
GPU Power Consumptions



- My electricity bill did go up by ~$100 for the month that I trained models non-stop…
- And my room temperature went up by 2 degrees
  - The corner where the computer sits was always warm

ACSIP
IT 资深人协会

Conclusion: Cloud vs Own



Nvidia GTX 1080 Ti vs AWS Tesla K80: ~4X

To gain performance of one GTX 1080TI equivalent in the cloud running 24x7 = $2500/ mth

ACSIP
IT 资深人协会

The Data Science Workflow

- Data Exploration
- Data Preprocessing, e.g. augmentation
- Data Splits
  - Training
  - Dev (Validation)
  - Test
- Decide Evaluation Metrics
- Training and Experiments (One Factor At A Time)

ACSIP
IT 资深人协会

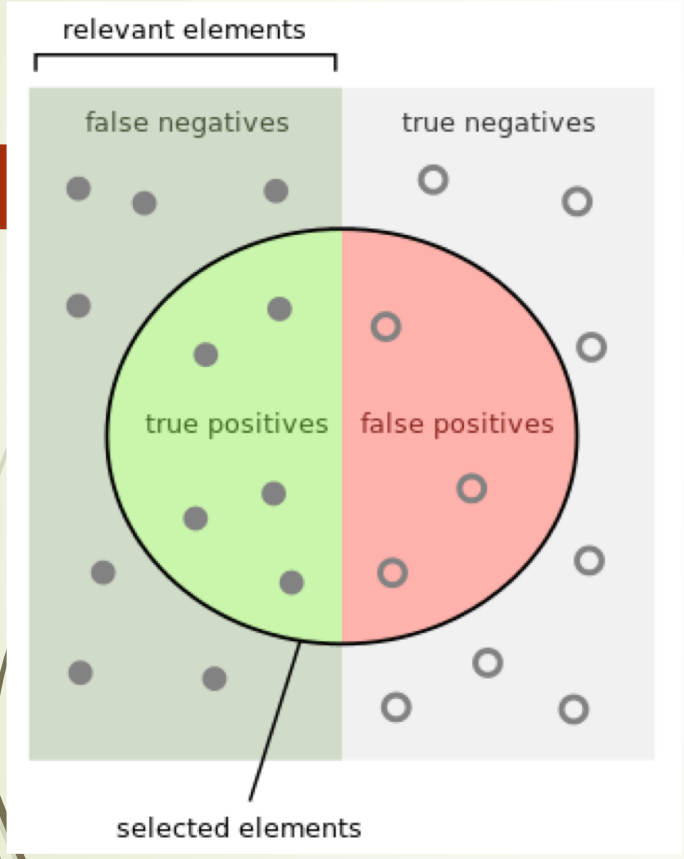
Is accuracy a good measurement

- A Disease: 1 out of 1 million probability
- A test: 99.99% accuracy
- If you are tested positive, should you be worried?

- Out of 1 million prediction, 100 are predicted wrong
- Out of 1 million population, there is only 1 person who has it

ACSIP
IT 资深人协会

Evaluation Matrix



How many selected items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{selected elements}}$$

When I say positive, chances of really positive

How many relevant items are selected?

$$\text{Recall} = \frac{\text{true positives}}{\text{relevant elements}}$$

Out of all the positive cases, how many did I get right

ACSIP
IT 资深人协会

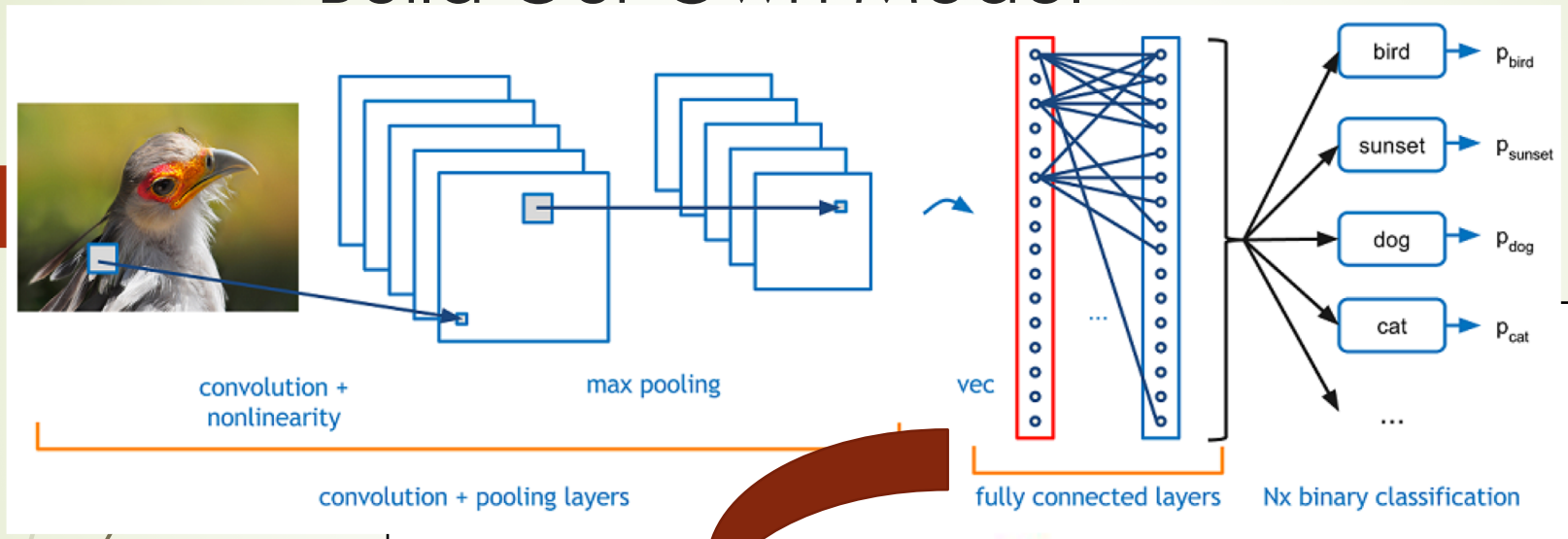
MURA Evaluation: Cohen's Kappa

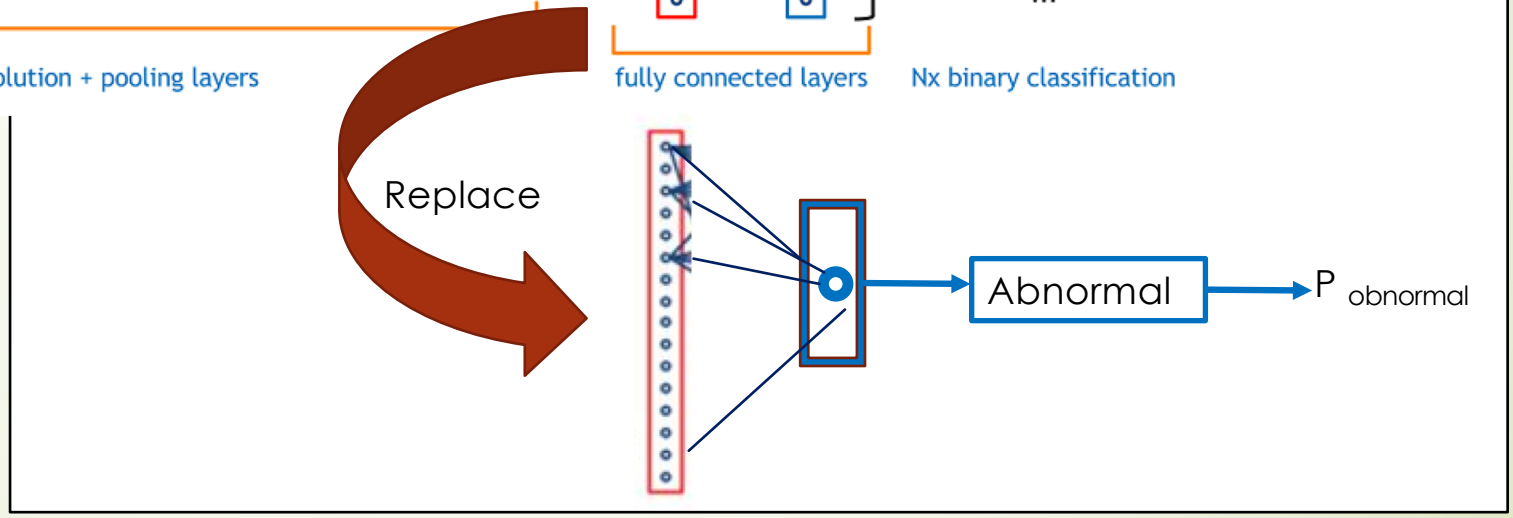
$$k = \frac{p_o - p_e}{1 - p_e}$$

- P_0: relative observed agreement among raters
- P_e: hypothetical probability of chance agreement

ACSIP
IT 资深人协会

Build Our Own Model





ACSIP
IT 资深人协会

Replacing the TOP

- vgg_conv = VGG16(weights='imagenet', **include_top=False**)
- model = models.Sequential().add(vgg_conv)
- model.add(layers.Flatten())
- model.add(layers.Dense(2, activation='softmax'))

ACSIP 资深人协会

Data Augmentation

- Horizontal Flipping
- Vertical Flipping
- Rotation
- Translation
- Changing brightness
- …
- …

Prevent Overfitting

ACSIP
IT 资深人协会

Tuning Hyper Parameters

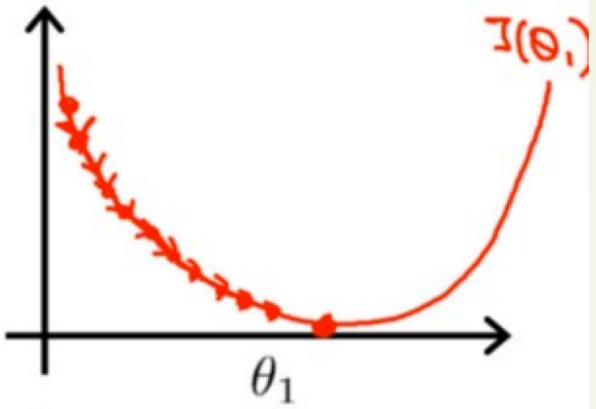
- Model Architecture
  - MobileNet, VGG, Densenet, Resnet … …
  - Multitasking
  - Body Part Specific Models
  - RNN Top
- Batch Size
- Learning Rate Decay, Cos Annealing, and Early Stopping
- Regularization
- Backprop Algorithm (SGD, Adam, Adagrad …)
- Learning Rate Optimization

ACSIP
IT 资深人协会

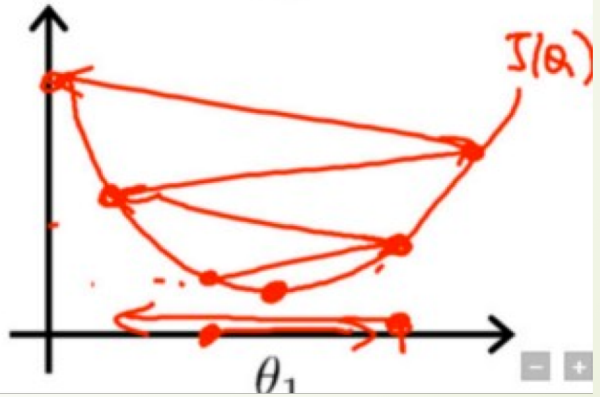
Training – Learning Rate

$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

If α is too small, gradient descent can be slow.

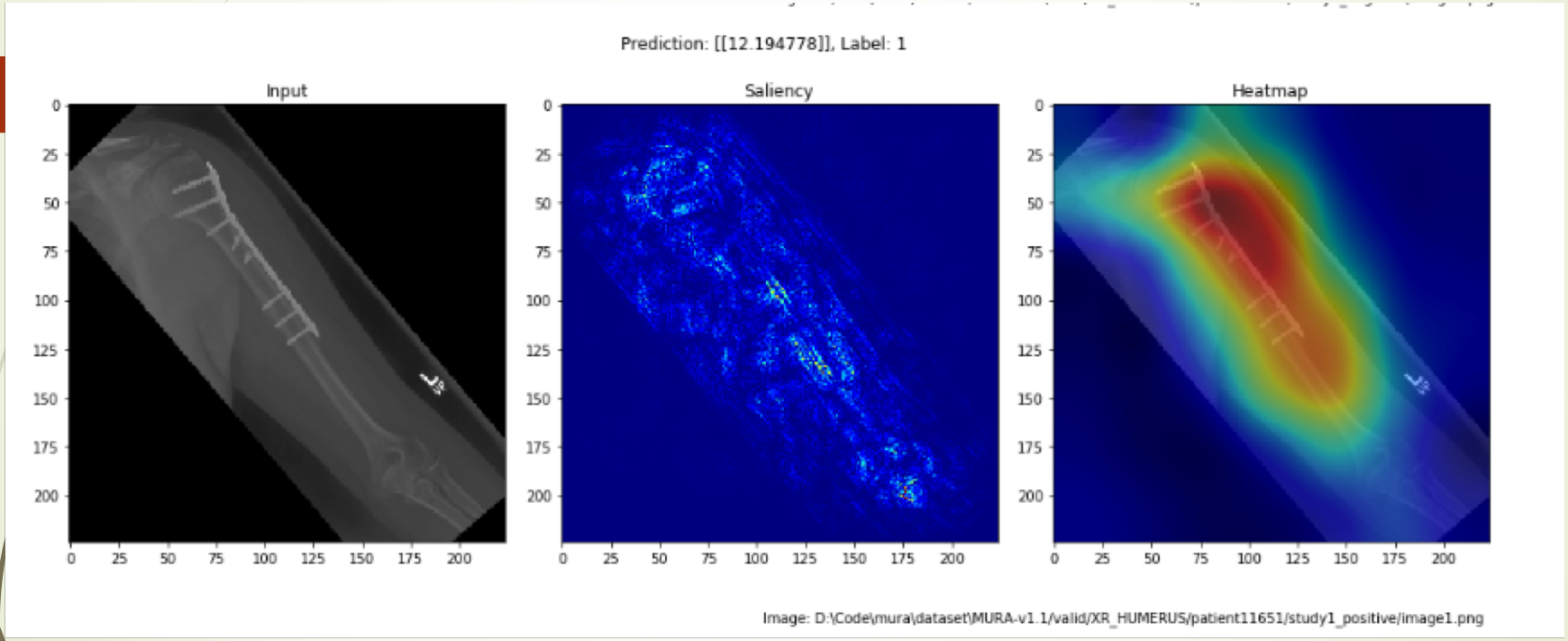


If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



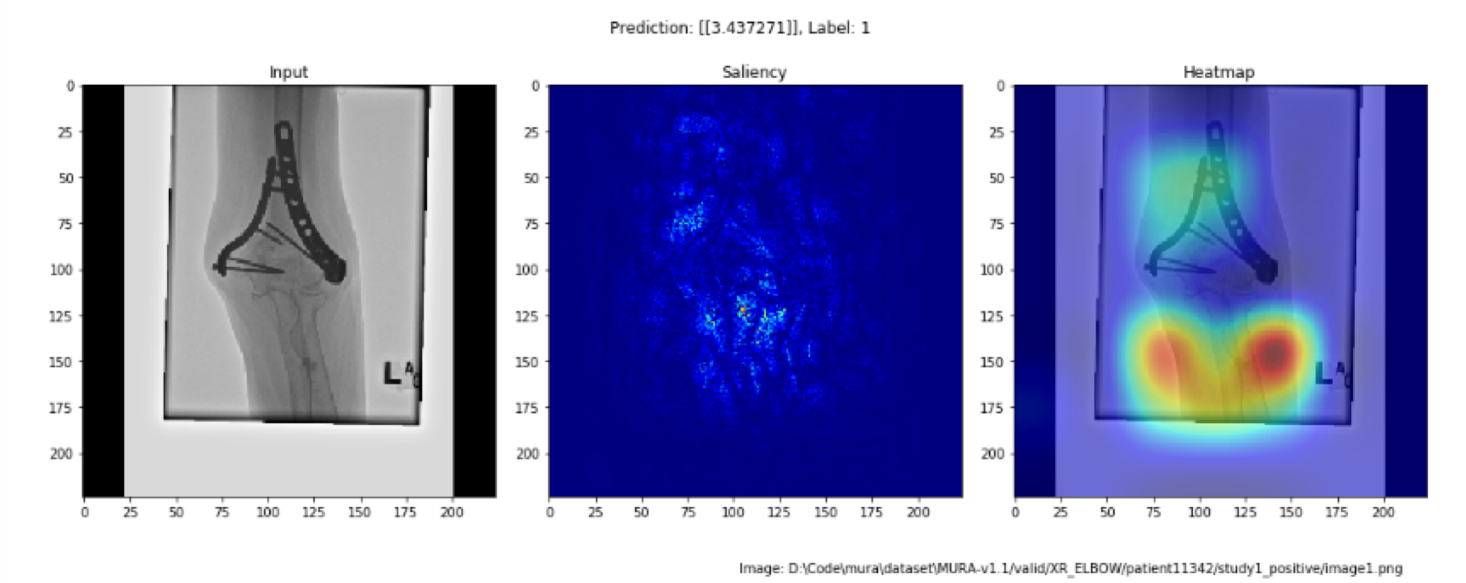
ACSIP 资深人协会

Visualization – Correct Reasoning



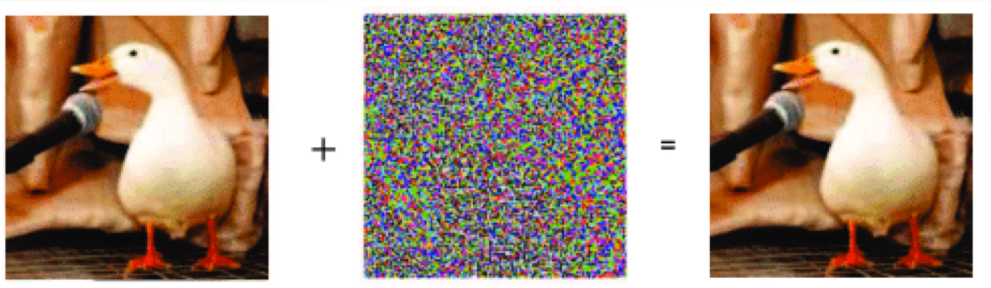
ACSIP
IT 资深人协会

Visualization – Wrong Reasoning



ACSIP
IT 资深人协会

Adversarial Attack



ACSIP 资深人协会

Submission – CodaLab



- Submit your code
- Run through some sample data
- Tag it
- They will run your model with real validation data

ACSIP
IT 资深人协会

Next Steps

- Taking Online Courses
  - Andrew Ng's Deep Learning specialization on Coursera
  - CS229, CS231n and CS224d from Stanford
  - Deep Reinforcement Learning from UC Berkeley
  - Fast.AI Deep Learning Course
- Attending Conferences
  - NIPS, ICML, ICLR, ACL, ReWork Deep Learning Submit
- Reading Papers
- Working On Projects (Competitions, Side Projects etc)

- MOST IMPORTANT: TIME COMMITMENT

ACSIP
IT 资深人协会

Future References

- http://forum.aisquaredforum.ca



- https://github.com/DeepMachineLearning/mura-team1
- https://github.com/DeepMachineLearning/mura-team2
- https://github.com/DeepMachineLearning/mura-team3