Deep Learning: Day 2

chyld @ galvanize

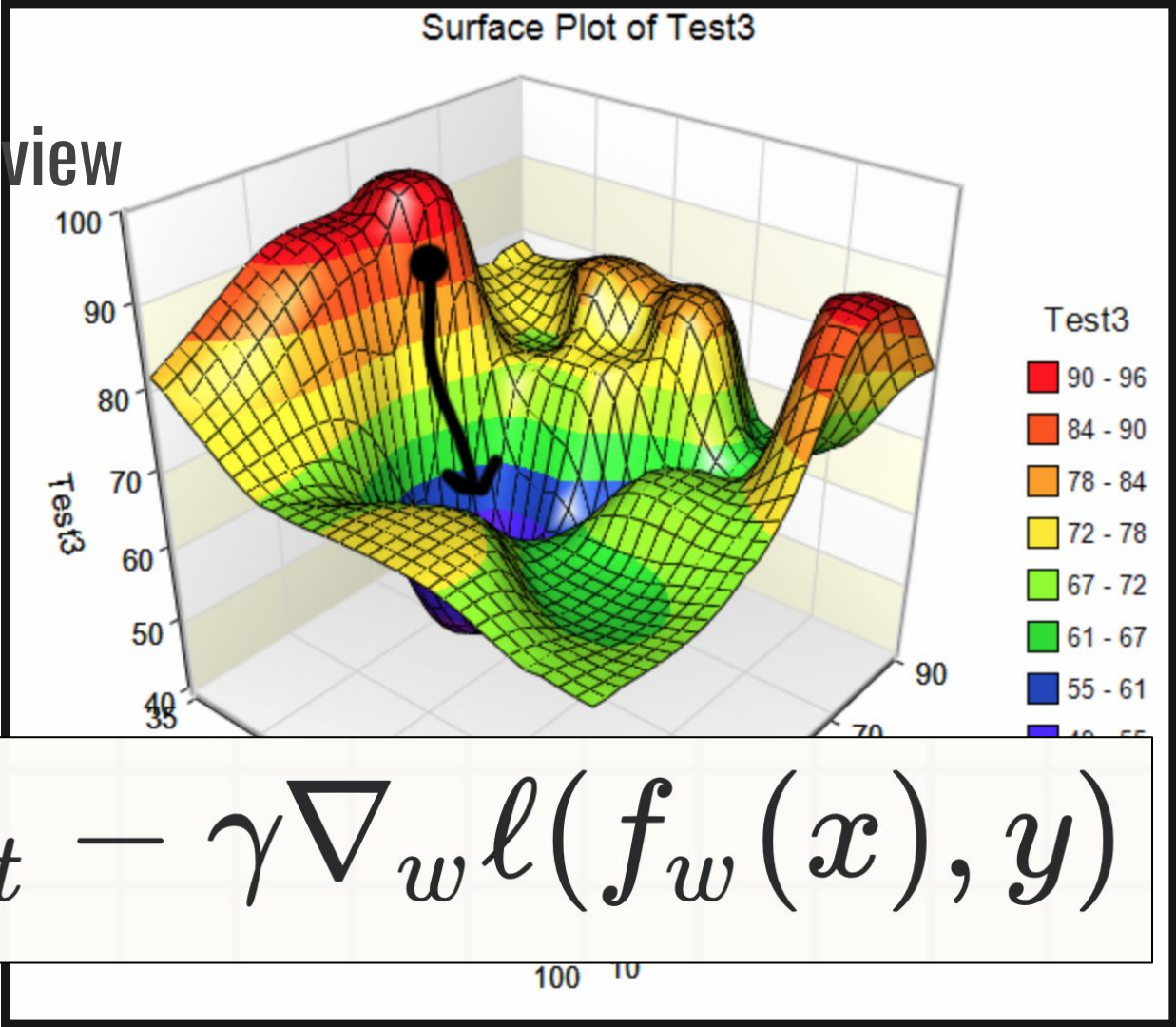
Topics

- Packages to Install
- Gradient Descent Review
- Tensorflow
- Keras
- Regression in Keras (Simple & Multiple)
- Convolutional Neural Networks
  - Digit Recognition
  - Clothing Recognition
  - Large Image Recognition
- Amazon Web Services
- Kaggle

Packages to Install

```
pip install -U keras tensorflow scikit-learn
```

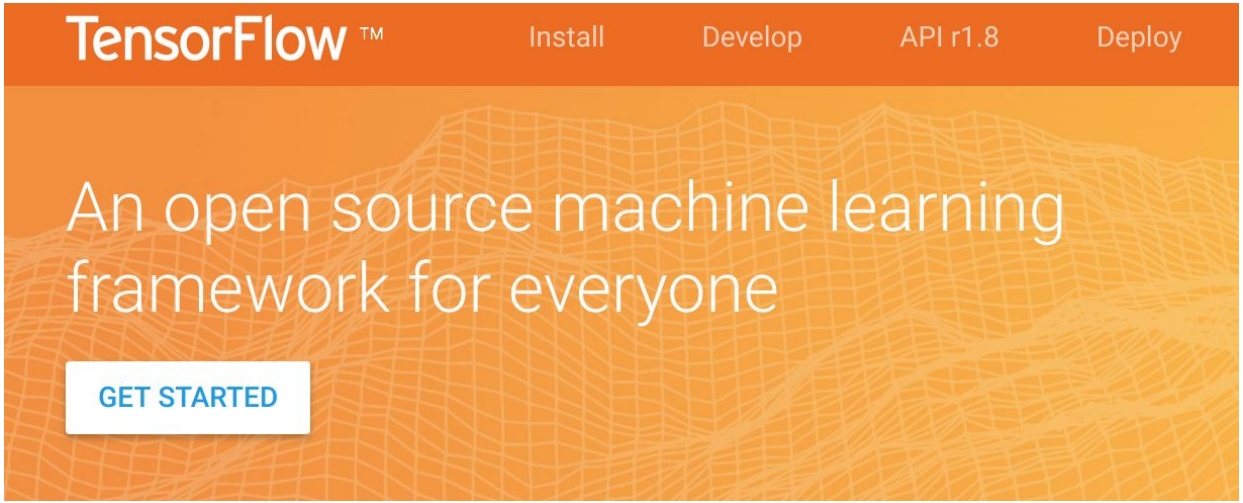
Gradient Descent Review



$$w_{t+1} = w_t - \gamma \nabla_w \ell(f_w(x), y)$$

Tensorflow

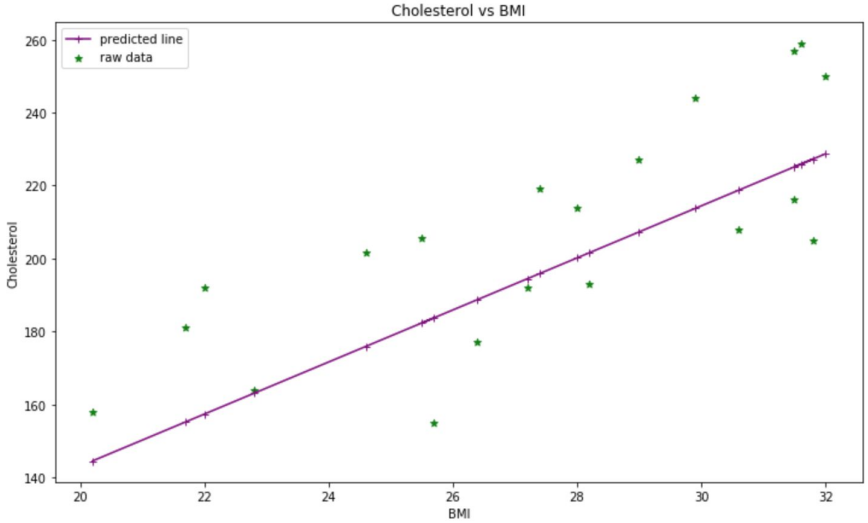
- https://www.tensorflow.org/
- An open source machine learning framework for everyone

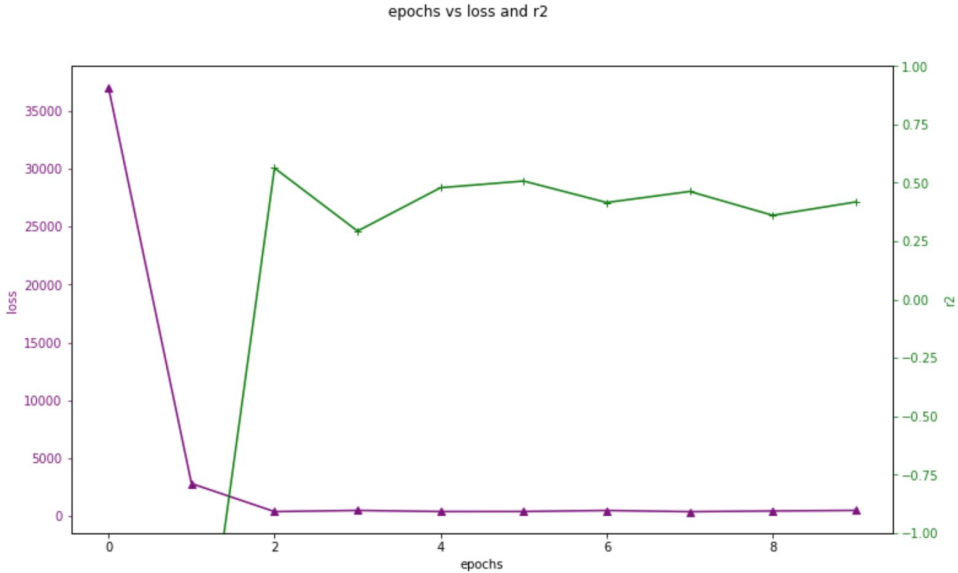


Keras

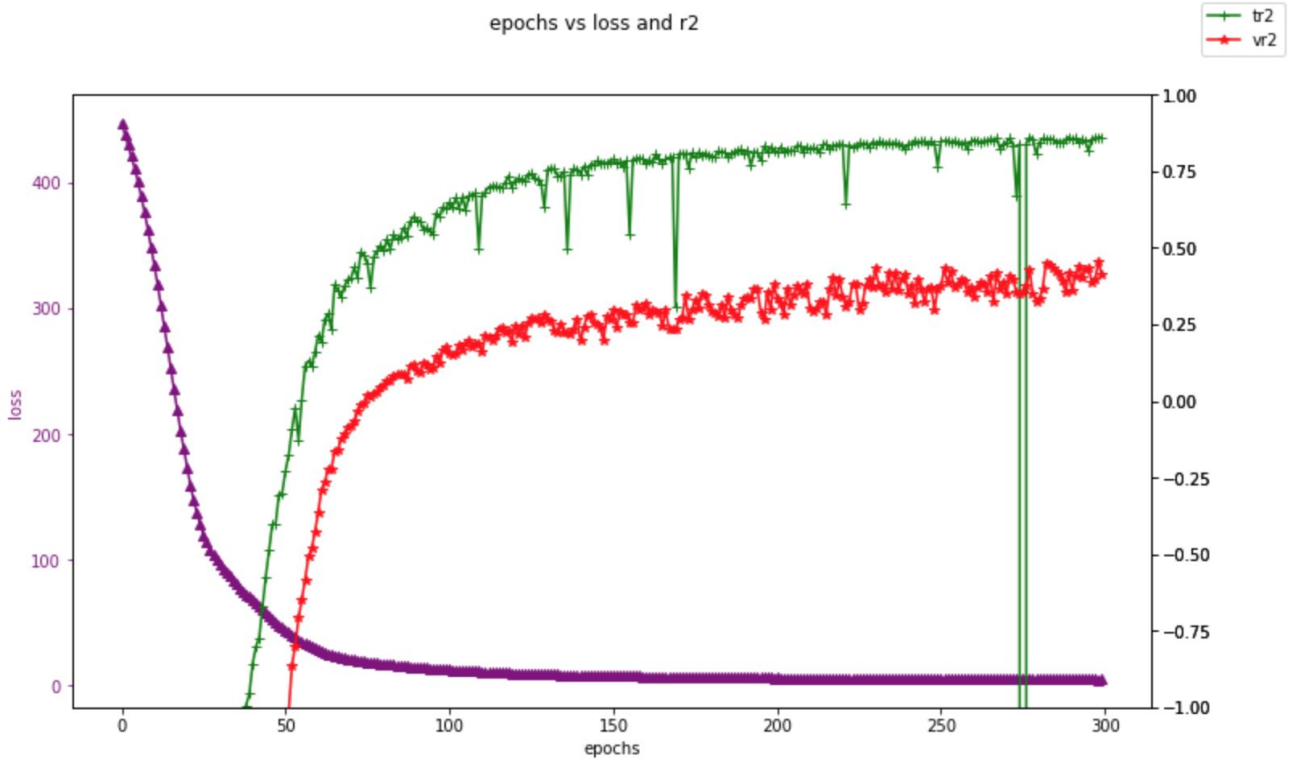
- https://keras.io/
- Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation.

Simple Linear Regression with Keras



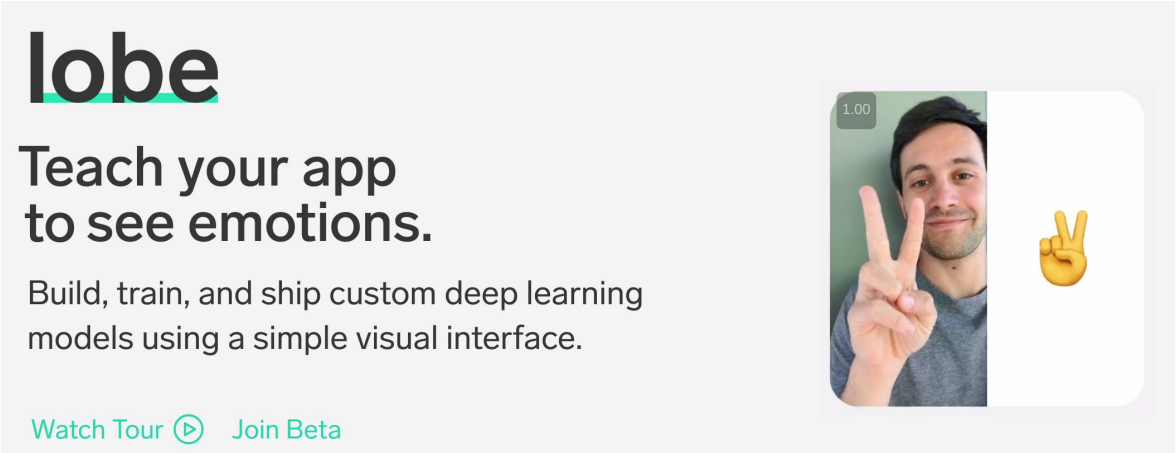


Multiple Linear Regression with Keras

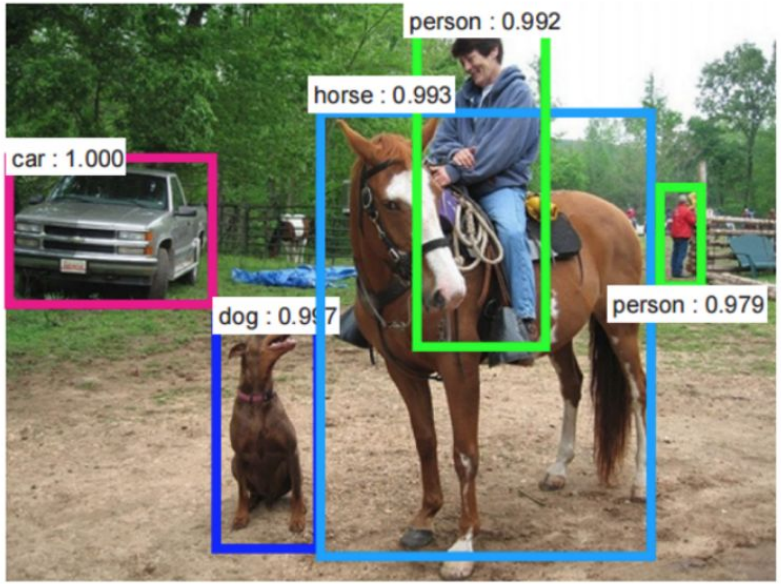


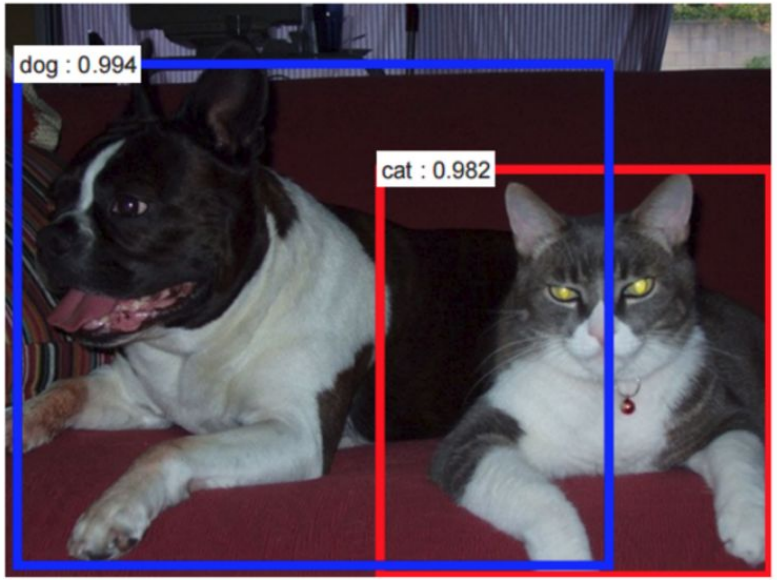
Applications of Convolutional Neural Networks

- https://lobe.ai/tour

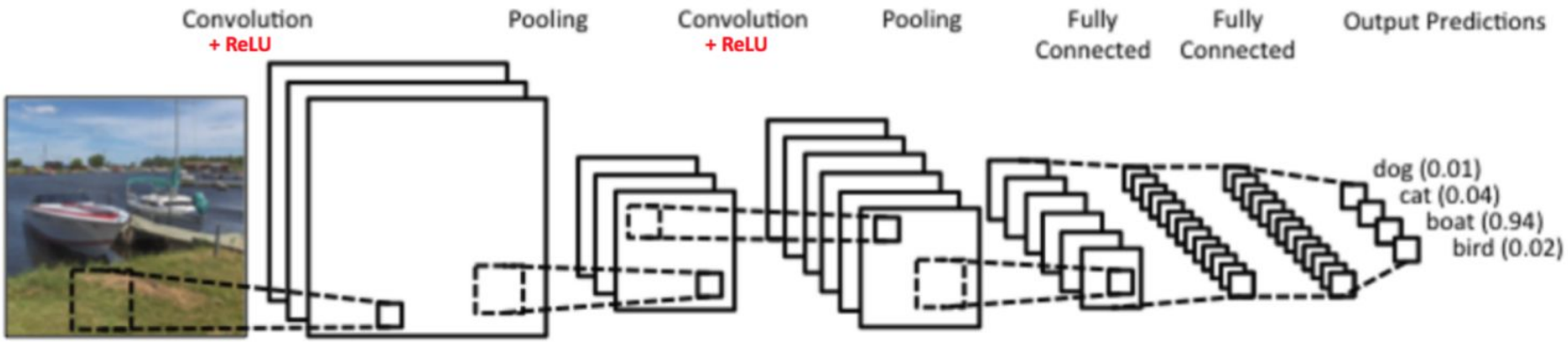


Convolutional Neural Networks (CNN)

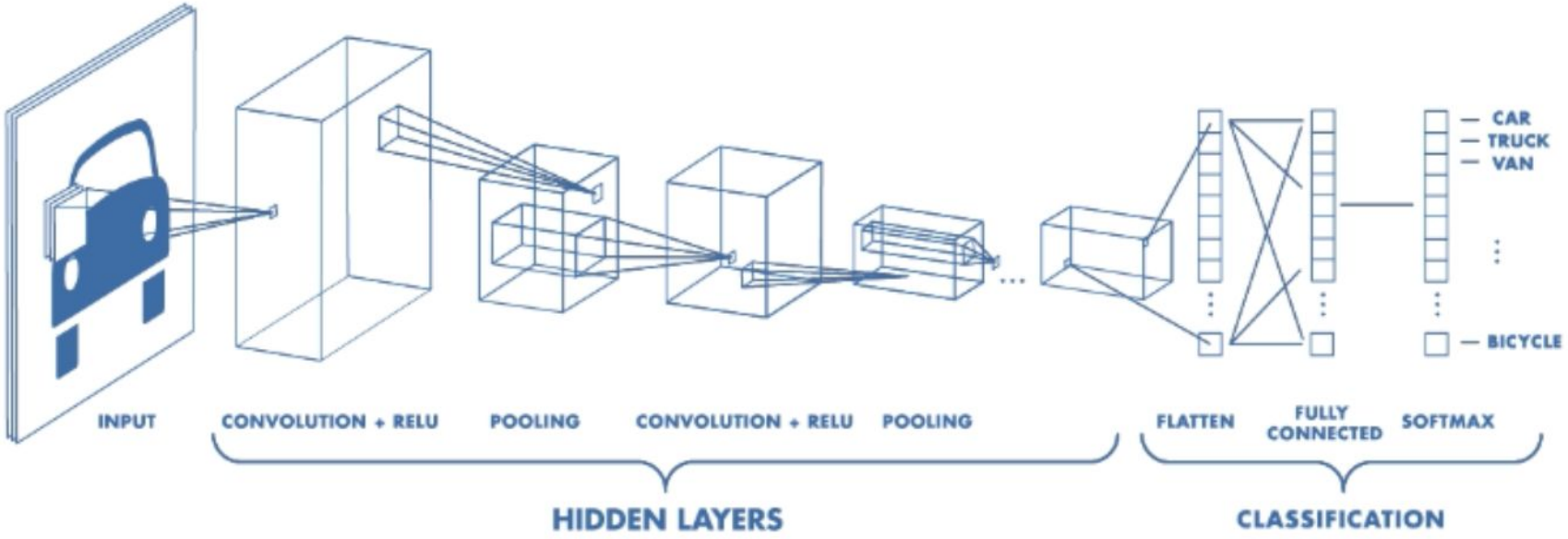




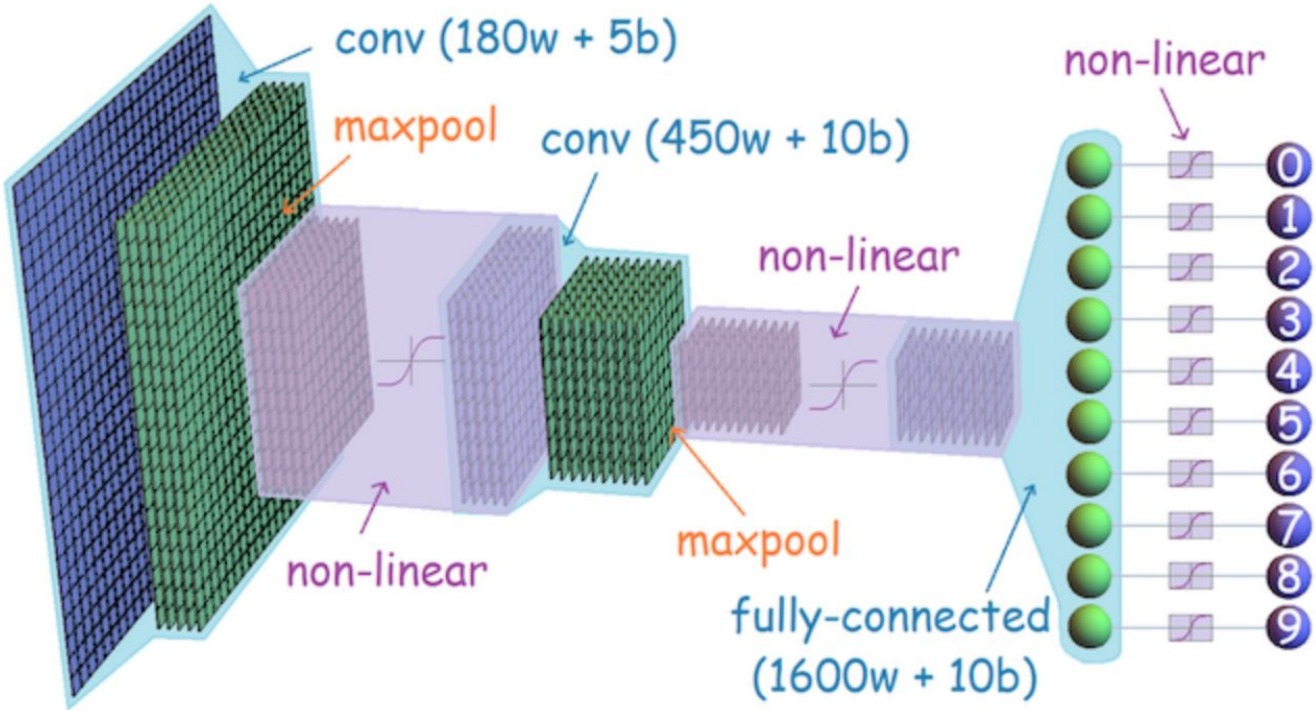
CNN Architecture



CNN Architecture

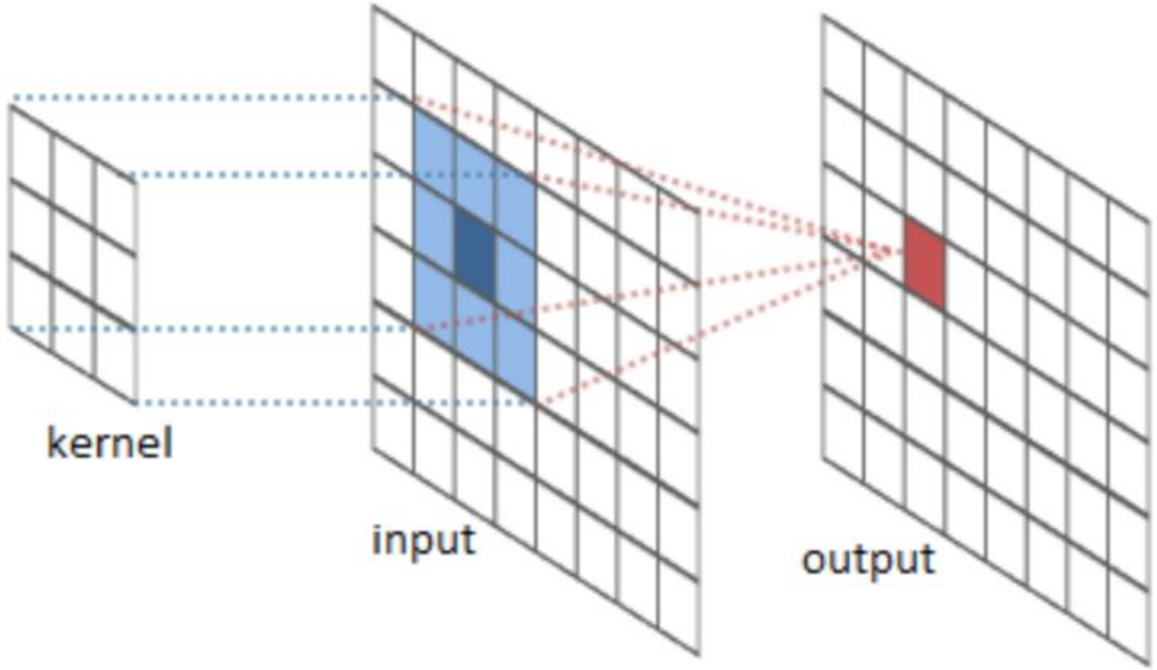


CNN Architecture



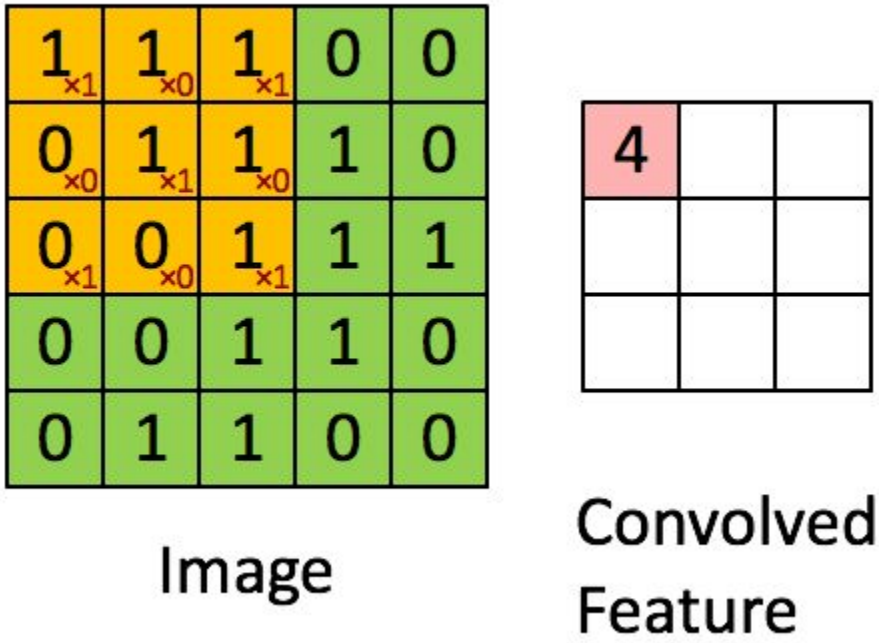
Filter convolving over an image creates feature map

Filter convolving over an image creates feature map

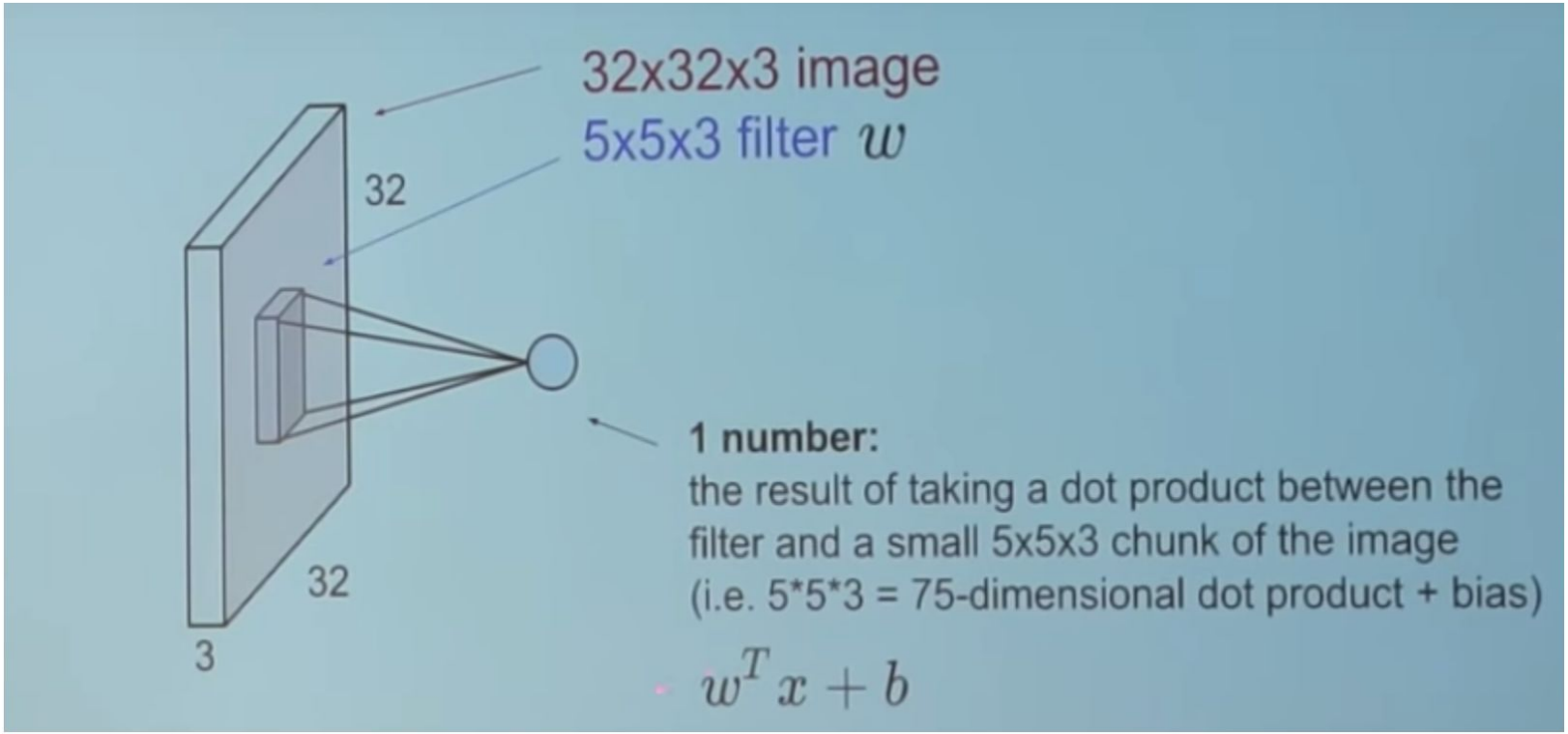


Padding & Stride

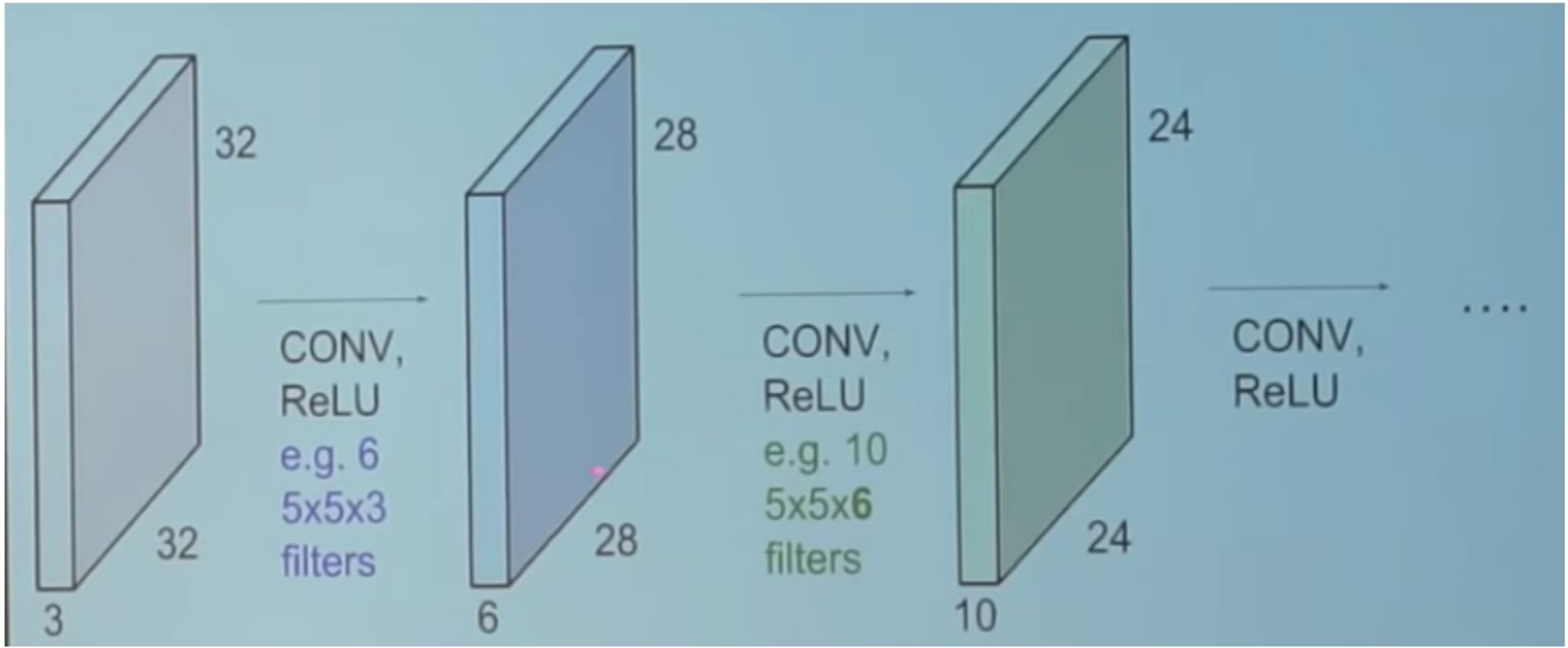
Sum of the products



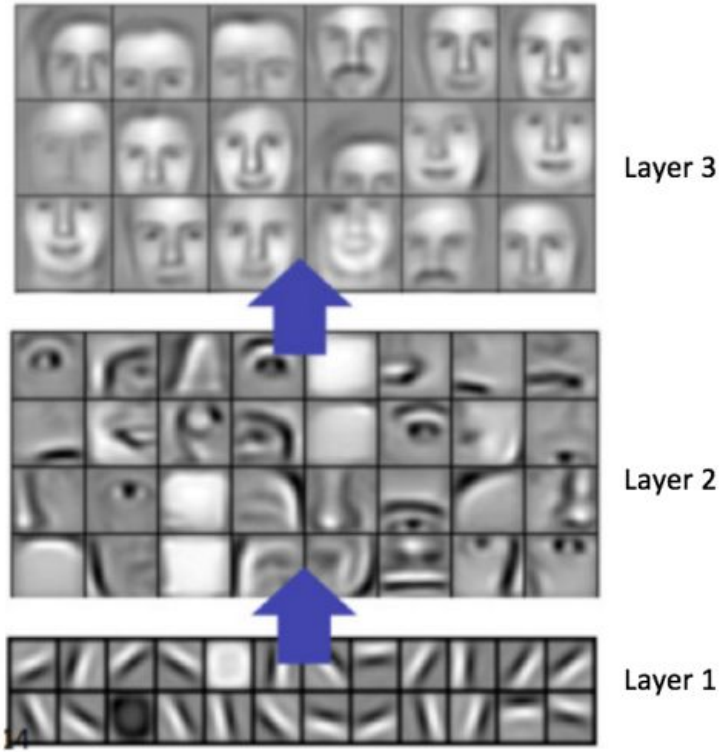
Dot Product


32x32x3 image
5x5x3 filter w
32
32
3
1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)
$w^T x + b$

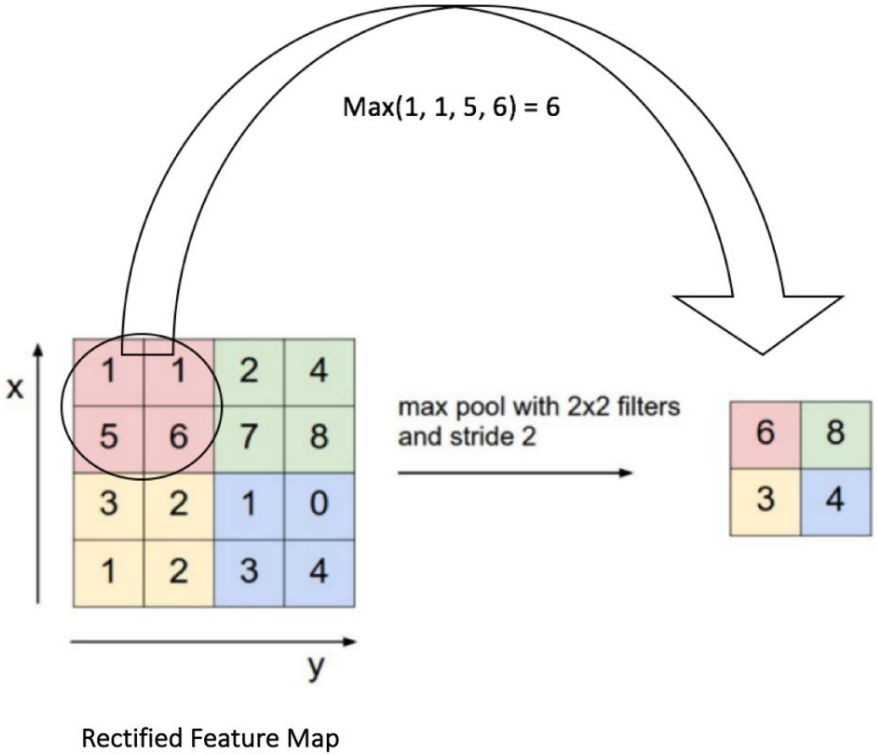
Stacking Convolutional Layers



Visualizing CNN Filters



Max Pooling Layer

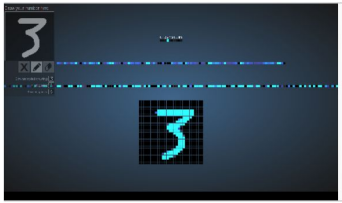


Interactive Network Visualizer

http://www.cs.cmu.edu/~aharley/vis/

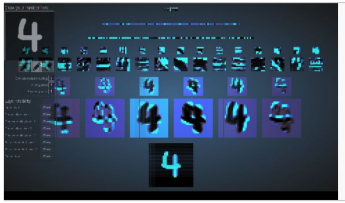
[3D convolutional network visualization]

This network has 1024 nodes on the bottom layer (corresponding to pixels), six 5x5 (stride 1) convolutional filters in the first hidden layer, followed by sixteen 5x5 (stride 1) convolutional filters in the second hidden layer, then three fully-connected layers, with 120 nodes in the first, 100 nodes in the second, and 10 nodes in the third. The convolutional layers are each followed by downsampling layer that does 2x2 max pooling (with stride 2).



[2D fully-connected network visualization]

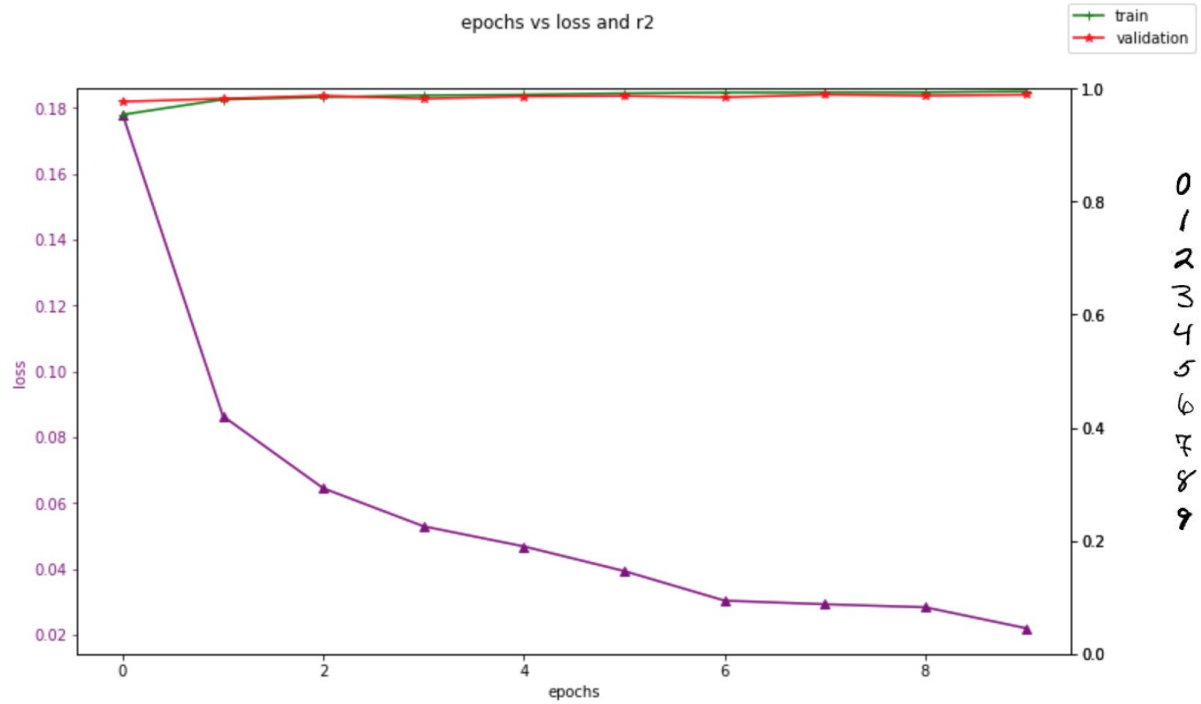
This is the same as the first visualization, but with the nodes flattened on a plane so that they are easier to see all at once.

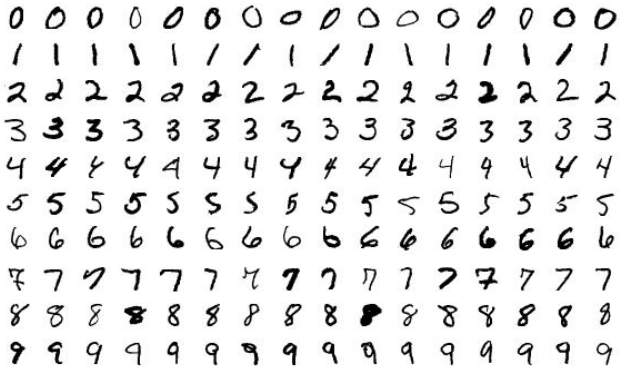


[2D convolutional network visualization]

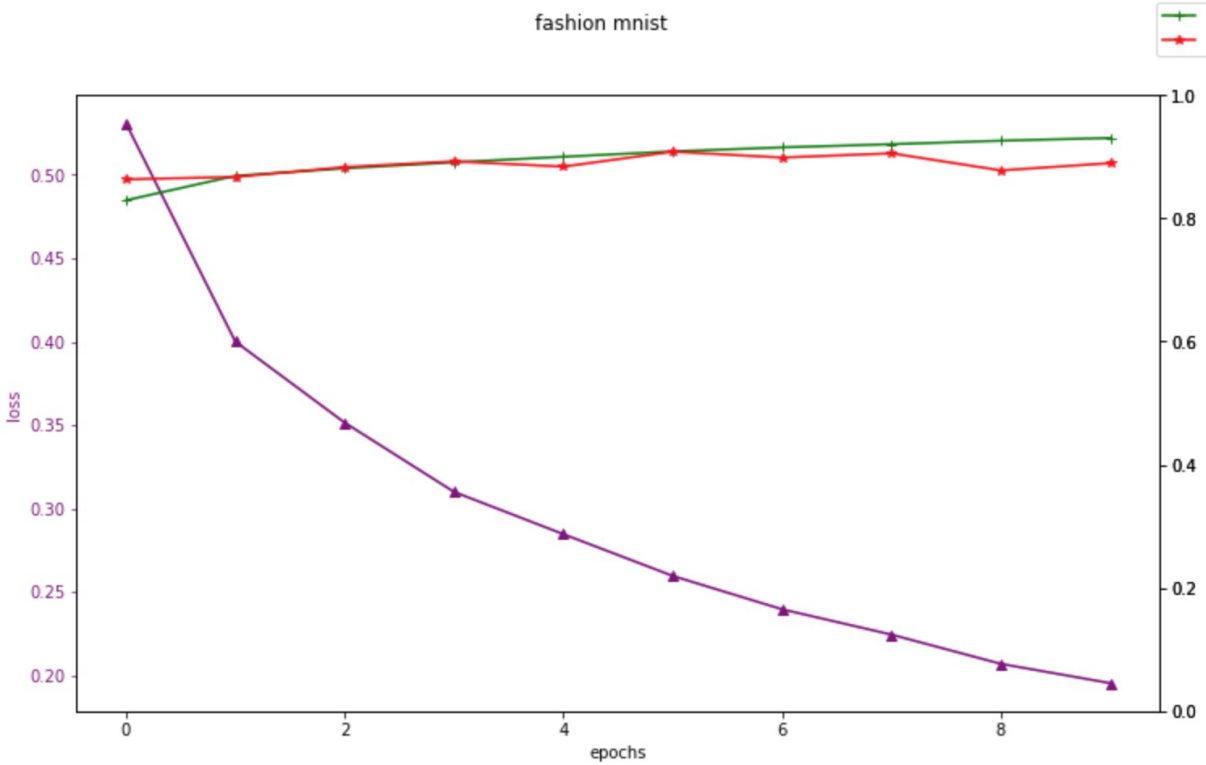
This is the same as the second visualization, but with the nodes flattened on a plane so that they are easier to see all at once.

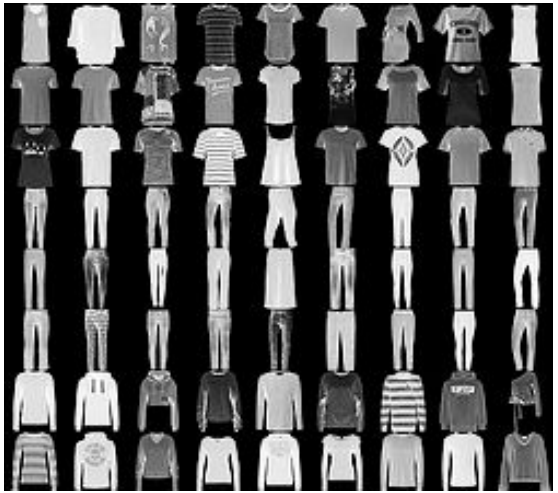
CNN: Handwritten Digit Recognition



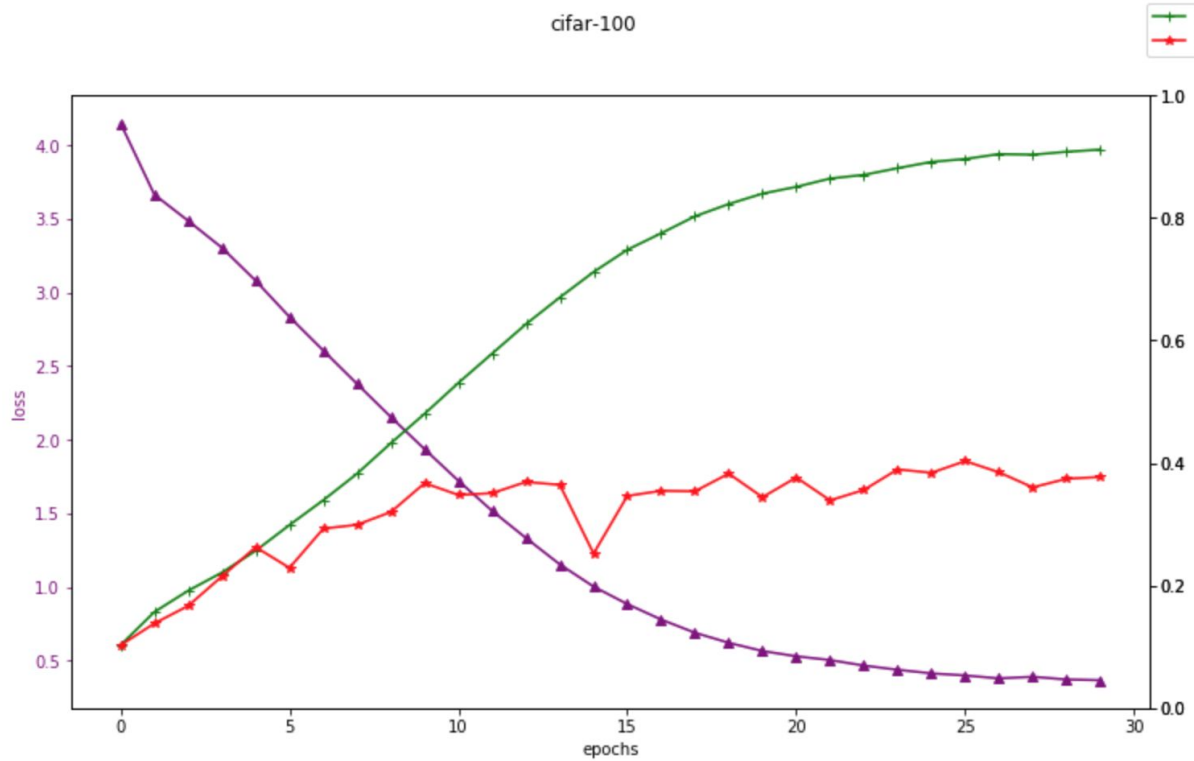


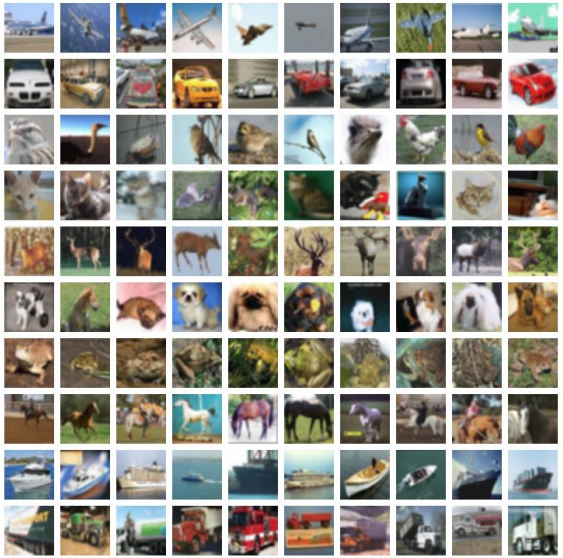
CNN: Clothing Recognition





CNN: CIFAR-100 Image Recognition





Amazon Web Services

Free tier eligible

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

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"