Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Deep Learning for Self-Driving Cars

Hamid D. Taghirad
Alireza Norouzzadeh

http://aras.kntu.ac.ir

November 6, 2017

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Contents

1 Introduction to Deep Neural Networks

2 Finding lane lines by low-level image processing techniques

3 Introduction to TensorFlow

4 Traffic sign classification by TensorFlow

5 Introduction to Keras

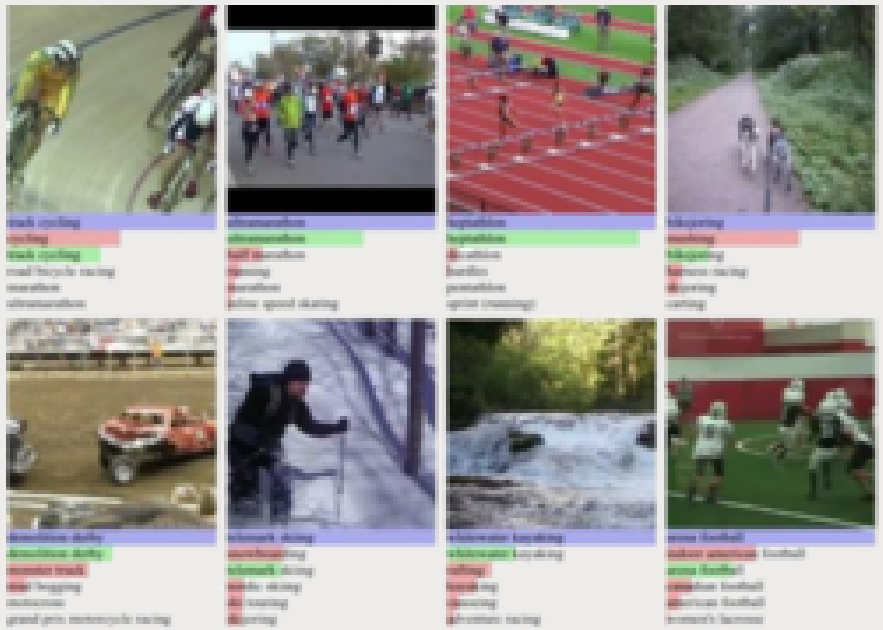
6 Behavioral cloning

7 Accurate and efficient vehicle detection

Introduction

Applications of Deep Learning

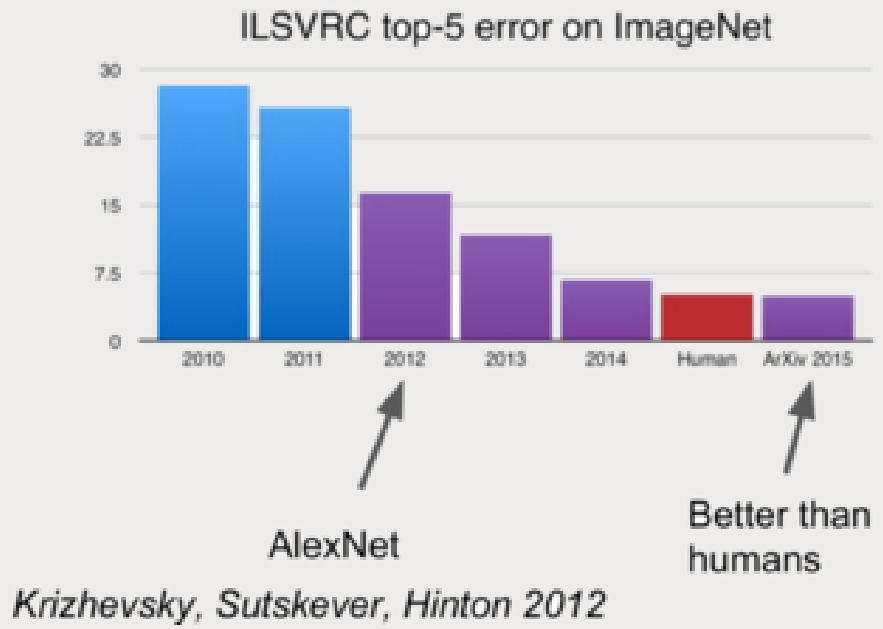
- Image Classification: Labeling an image.



Introduction

Applications of Deep Learning

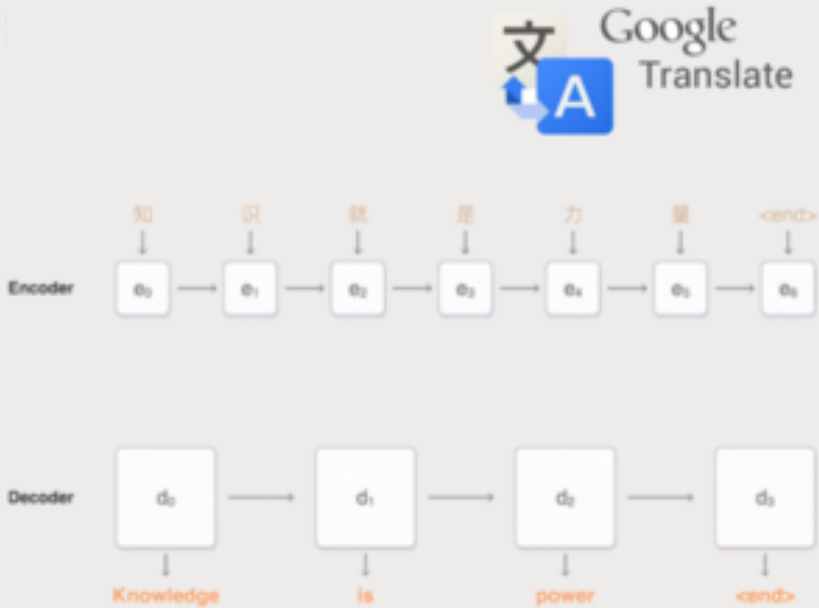
- Image Classification: Labeling an image.



Introduction

Applications of Deep Learning

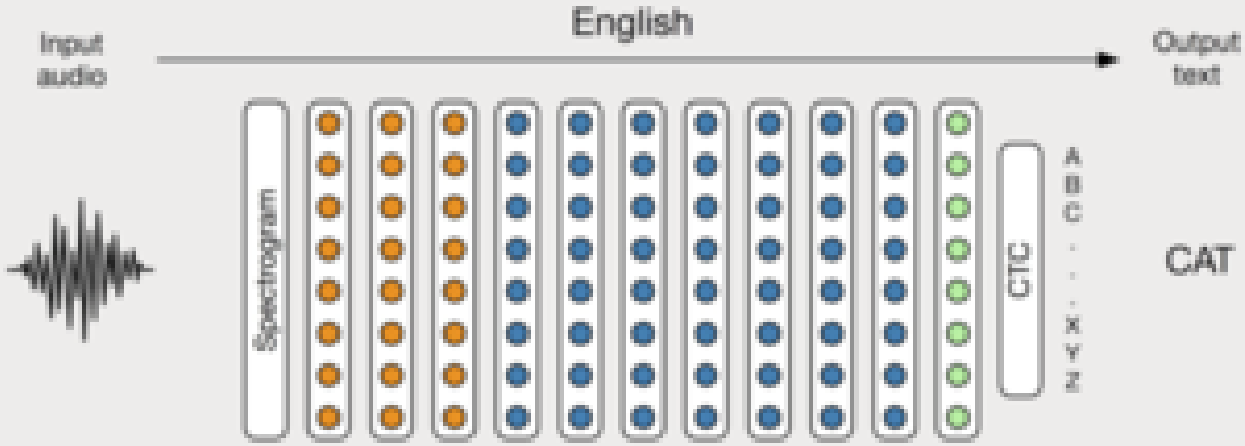
- Machine Translation: Translating from one language to another.



Introduction

Applications of Deep Learning

- Speech Recognition: Conversion of speech to text.



Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction

Applications of Deep Learning

- Game Playing: Deep Mind.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

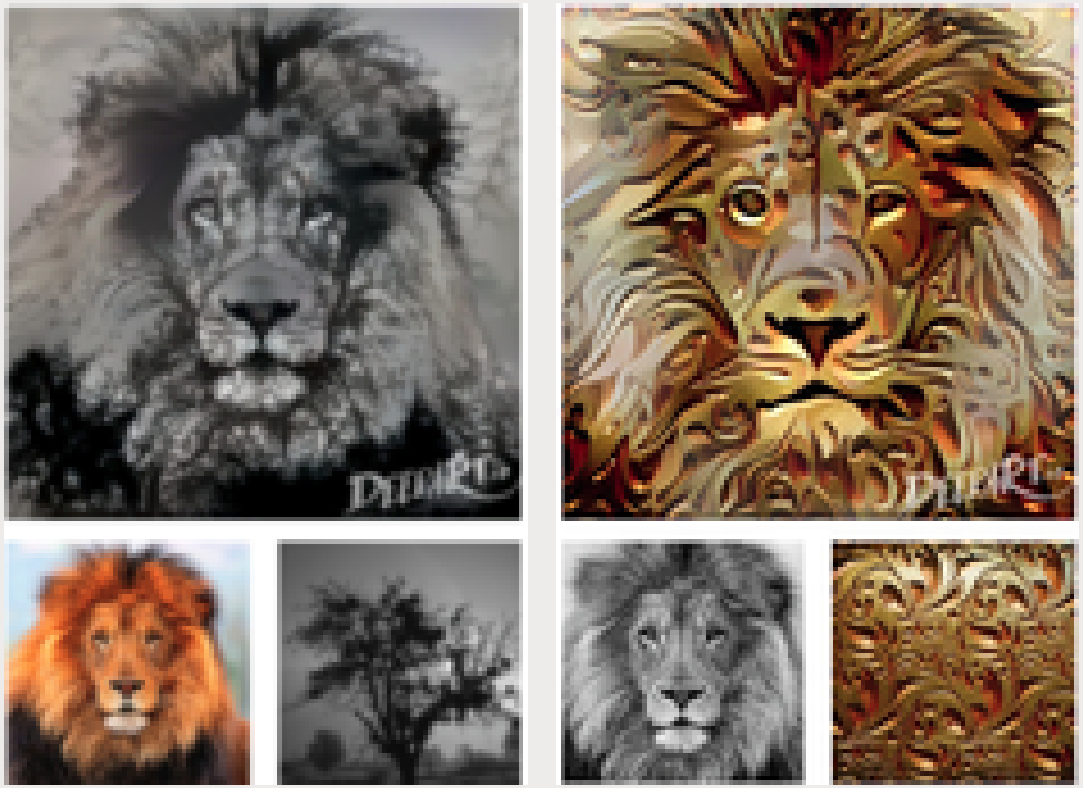
Behavioral cloning

Accurate and efficient vehicle detection

Applications of Deep Learning

Style transfer

- http://deepart.io/



Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

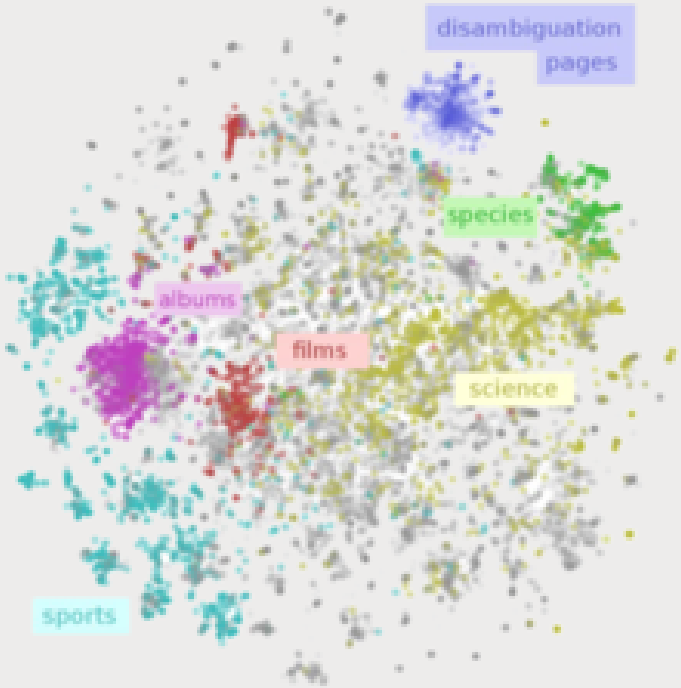
Behavioral cloning

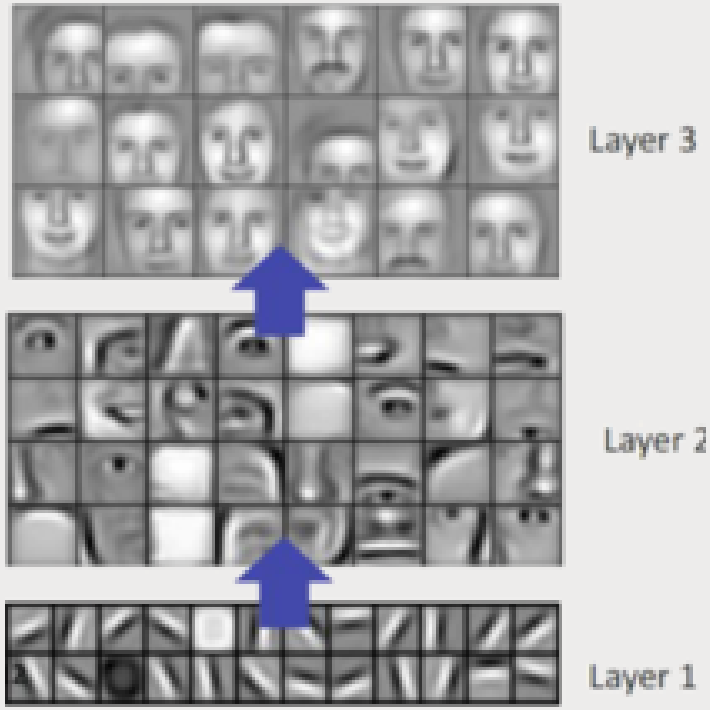
Accurate and efficient vehicle detection

Introduction

Why Deep Learning?

- Hand-Engineered Features vs. Leanred Features.





Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

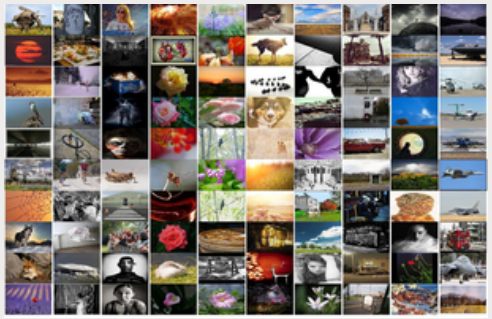
Behavioral cloning

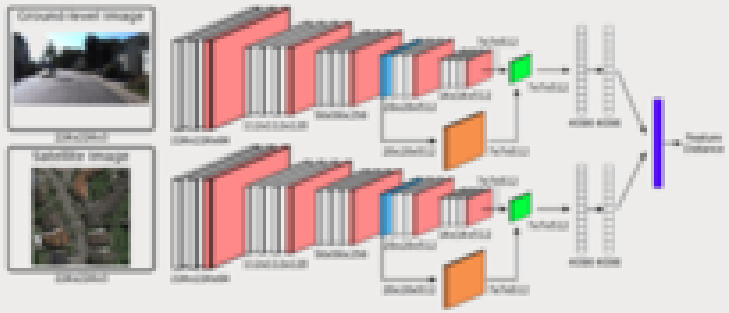
Accurate and efficient vehicle detection

Introduction

Why Now?

- Large Datasets.
- GPU Hardware Advances + Price Decreases
- Improved Techniques

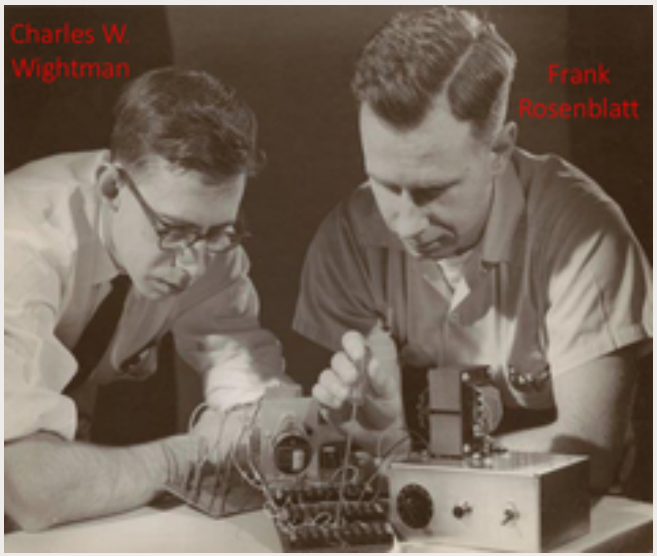


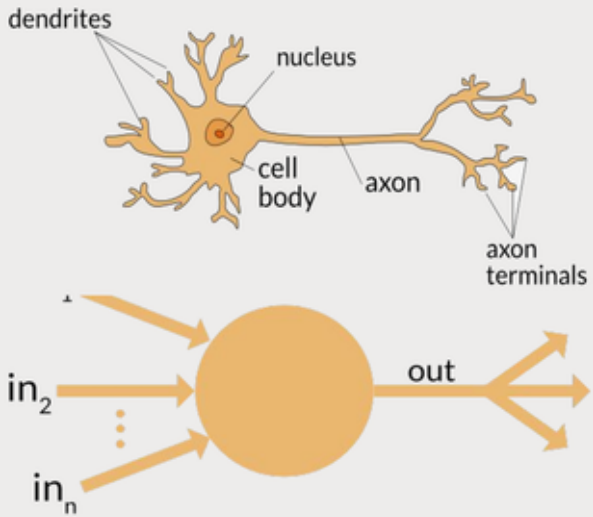


Introduction

The Perceptron

- Invented in 1954 by Frank Rosenblatt.
- Inspired by neurobiology.

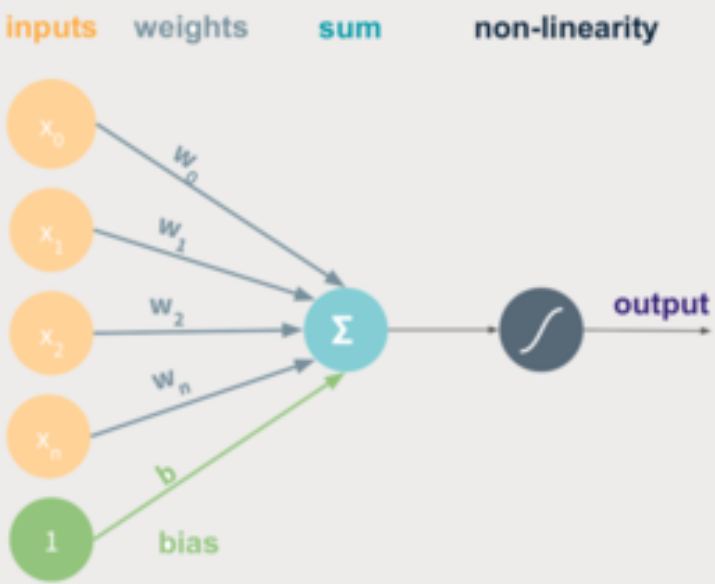




Introduction

The Perceptron Forward Pass

- Inputs are multiplied by weights.
- A bias is added.
- The activation function is applied.



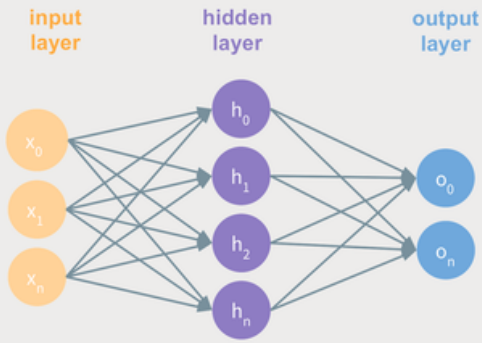
$$output = g(XW + b)$$

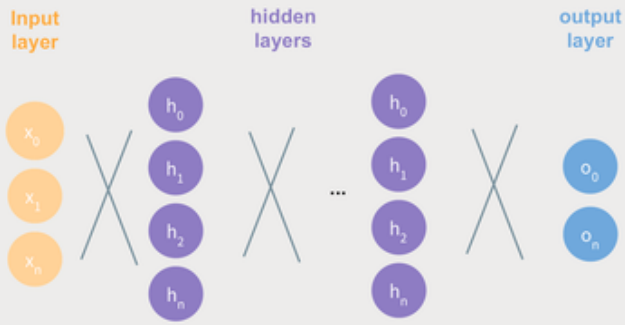
$$g(a) = \sigma(a) = \frac{1}{1 + e^{-a}}$$

Introduction

Multi-Layer & Deep Neural Networks

- A perceptron can have more than one output.
- A multi-layer perceptron has at least one hidden layer.
- Adding more hidden layers results in a deep neural network.





Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction

Deep Convolutional Neural Networks (CNNs)

- CNNs have repetitive blocks of neurons that are applied across space (for images) or time (for audio signals etc).
- For images, these blocks of neurons can be interpreted as 2D convolutional kernels.
- For speech, they can be seen as the 1D convolutional kernels applied across time-windows.
- At training time, the weights for these repeated blocks are shared.
- The reason for choosing this special structure, is to exploit spatial or temporal invariance in recognition.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction

Regular neural networks vs. CNNs

- Convolutional neural networks (CNNs) are a biologically-inspired variation of the multilayer perceptrons (MLPs).
- Neurons in CNNs share weights unlike in MLPs where each neuron has a separate weight vector.
- Convolutional neural networks employ a weight sharing strategy that leads to a significant reduction in the number of parameters that have to be learned.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

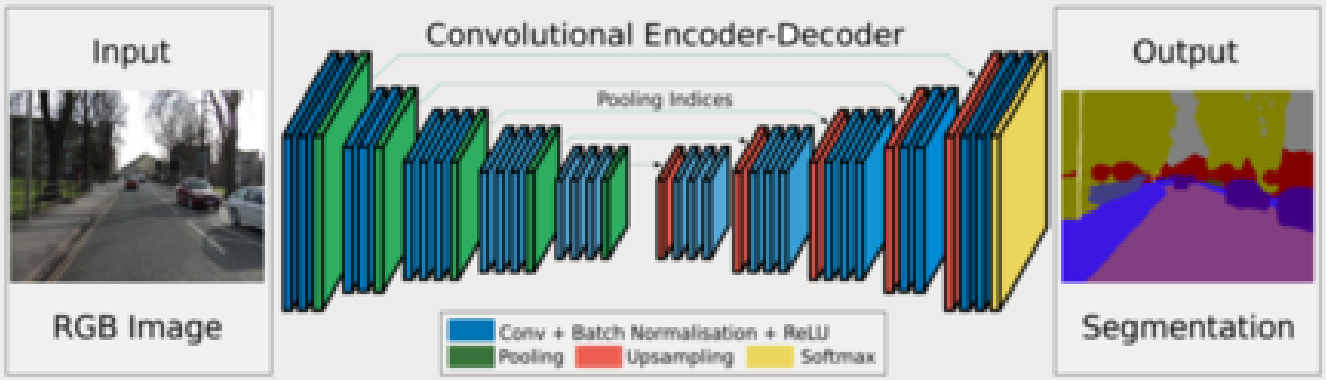
Behavioral cloning

Accurate and efficient vehicle detection

Introduction

Regular neural networks vs. CNNs

- We can think of deep learning as algorithms for automatic feature engineering.
- CNNs help us to overcome the vanishing gradient challenge and facilitate the learning in neural networks with many layers.



Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction

Deep Learning FrameWorks

- Caffe
- TensorFlow
- Microsoft CNTK
- Keras
- Theano
- Torch
- MXNet
- Deeplearning4j

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction

Caffe

- Interface: C++, Python
- One of the earliest GPU supported
- Initial focus on computer vision (and CNNs)

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction

TensorFlow

- Interface: Python, (C++)
- Automatic Differentiation
- Multi GPU, Cluster Support
- Currently most popular

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

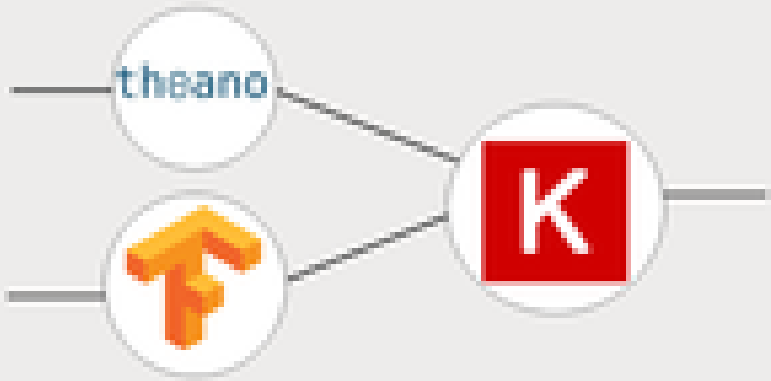
Behavioral cloning

Accurate and efficient vehicle detection

Introduction

Keras

- On top of Tensorflow (and Theano)
- Interface: Python
- Goal: provide a simplified interface
- Also: TF Learn, TF Slim



Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction

Theano

- Interface: Python (tight NumPy integration)
- One of the earlier frameworks with GPU support
- Encourages low-level tinkering

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

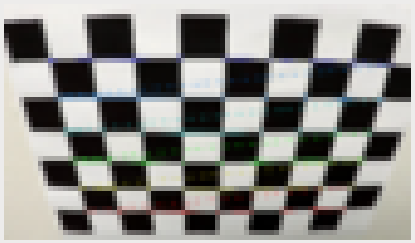
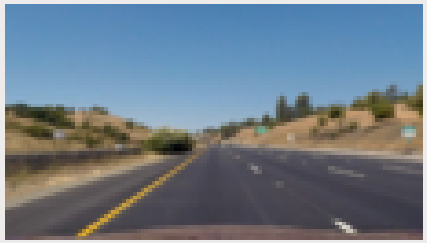
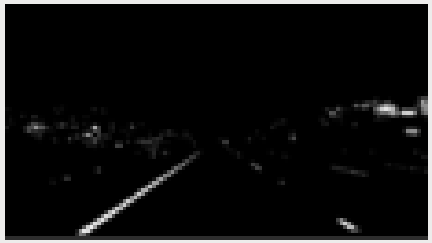
Behavioral cloning

Accurate and efficient vehicle detection

Finding Lane Lines

Algorithm overview

- Camera calibration
- Image thresholding
- Perspective transform
- Finding lines and drawing lane area

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

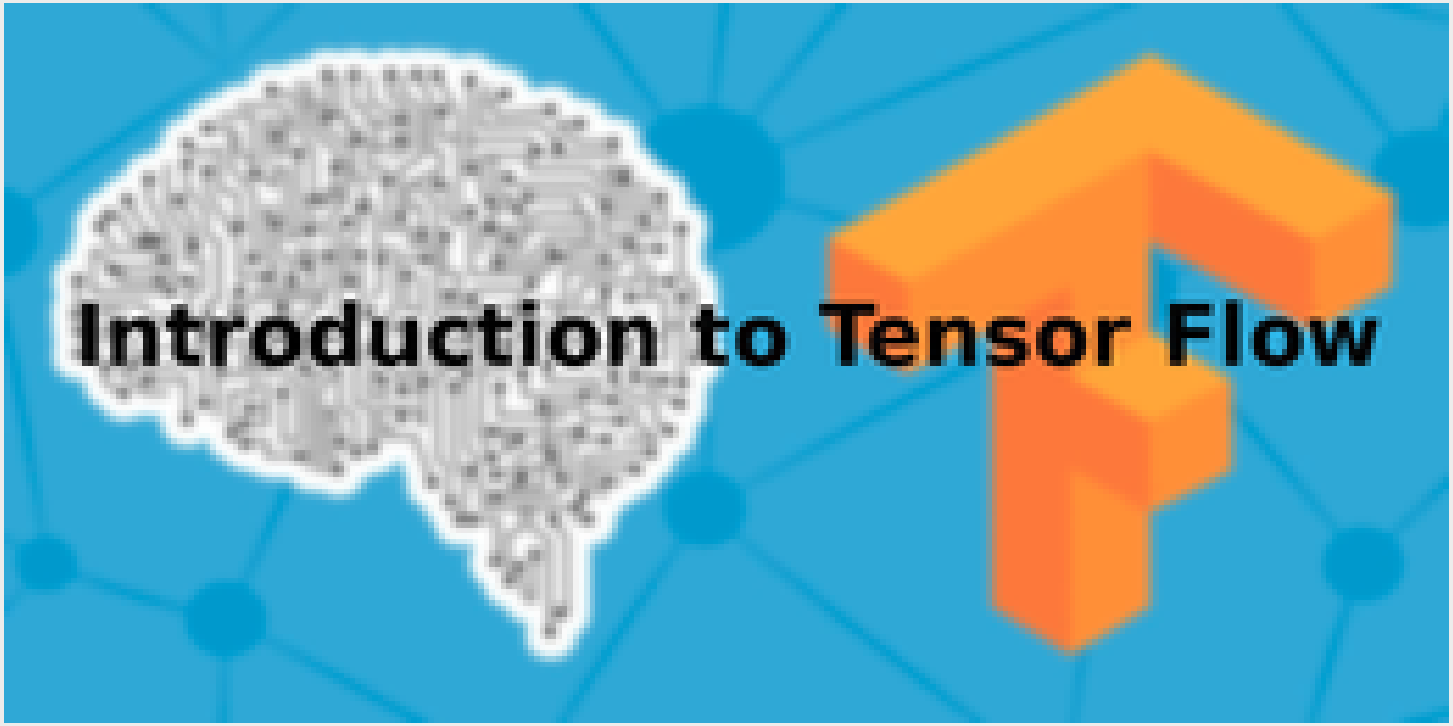
Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection



Introduction to Tensorflow

Tensorflow

- In TensorFlow, data is not stored as integers, floats, or strings.
- These values are encapsulated in an object called a tensor.
- There are some tensor types.
  - Variable
  - Constant
  - Placeholder
  - SparseTensor
- A TensorFlow session is an environment for running a graph on CPU/GPU.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction to Tensorflow

Tensorflow

- In TensorFlow, data is not stored as integers, floats, or strings.
- These values are encapsulated in an object called a tensor.
- There are some tensor types.
  - Variable
  - Constant
  - Placeholder
  - SparseTensor
- A TensorFlow session is an environment for running a graph on CPU/GPU.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction to Tensorflow

Hello world example

```
import tensorflow as tf

# Create TensorFlow object
hello_constant = tf.constant('Hello World!')

with tf.Session() as sess:
# Run the tf.constant operation
    output = sess.run(hello_constant)
    print(output)
```

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction to Tensorflow

Tensor types

```
# This is a 0-dimensional string tensor
hello_constant = tf.constant('Hello World!')
# A is a 0-dimensional int32 tensor
A = tf.constant(1234)
# B is a 1-dimensional int32 tensor
B = tf.constant([123,456,789])
# C is a 2-dimensional int32 tensor
C = tf.constant([[123,456,789],[222,333,444]])
```

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction to Tensorflow

Feeding Data into Tensorflow

```
x = tf.placeholder(tf.string)

with tf.Session() as sess:
    output = sess.run(x,
    feed_dict={x: 'Hello World'})
```

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction to Tensorflow

Feeding Data into Tensorflow

```
x = tf.placeholder(tf.string)
y = tf.placeholder(tf.int32)
z = tf.placeholder(tf.float32)

with tf.Session() as sess:
    output = sess.run(x,
    feed_dict={x: 'Test String',
    y: 123, z: 45.67})
```

Introduction to Tensorflow

Tensorflow Math

- Addition
- Subtraction
- Multiplication

```
x = tf.add(5, 2) # 7
x = tf.subtract(10, 4) # 6
y = tf.multiply(2, 5) # 10
```

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction to Tensorflow

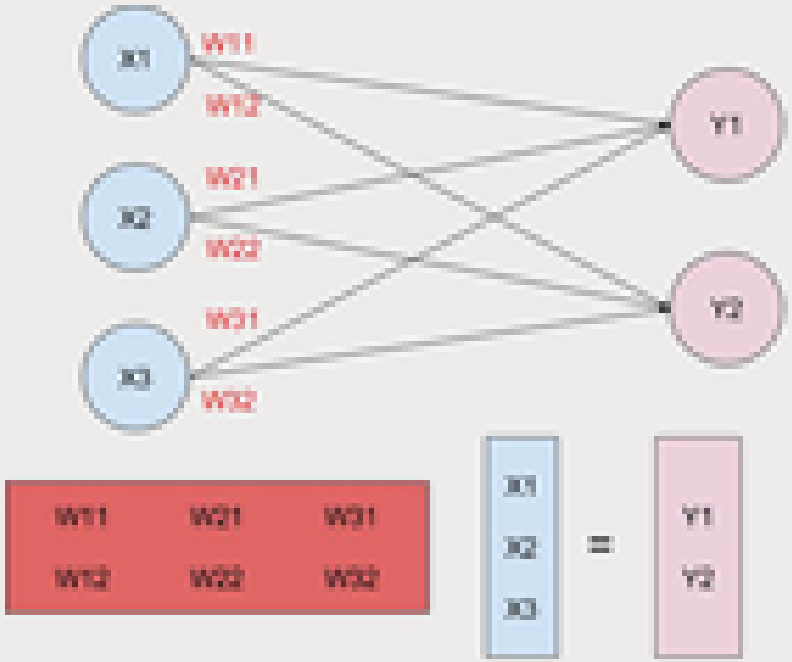
Converting Types

```
# Converting float to int32
tf.cast(tf.constant(2.0), tf.int32)
```

Introduction to Tensorflow

TensorFlow Linear Function

- Lets derive the function $y = Wx + b$.
- tf.placeholder() and tf.constant() can not be modified.
- The tf.Variable class creates a tensor with an initial value that can be modified.



Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction to Tensorflow

Weights and Bias in TensorFlow

- tf.global_variables_initializer() function is used to initialize the state of all the variable tensors.
- tf.truncated_normal() function is used to generate random numbers from a normal distribution.
- The tf.zeros() function returns a tensor with all zeros which is used for biases.

Introduction to Tensorflow

Weights

```
n_features = 120
n_labels = 5
weights = tf.Variable(tf.truncated_normal(
(n_features, n_labels)))
```

Introduction to Tensorflow

Biases

```
n_labels = 5
bias = tf.Variable(tf.zeros(n_labels))
```

Introduction to Tensorflow

Mnist Example

- get_weights function returns a tf.variable of weights.
- get_biases function returns a tf.variable of biases.
- The linear function $XW + b$ is implemented.
- All weights are initialized.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction to Tensorflow

Softmax

- The Softmax function is used to convert logits to probabilities.

```
def run():
    output = None
    logit_data = [20.0, 10.0, 1.0]
    logits = tf.placeholder(tf.float32)
    softmax = tf.nn.softmax(logits)

    with tf.Session() as sess:
        output = sess.run(softmax,
        feed_dict={logits: logit_data})
    return output
```

Introduction to Tensorflow

Cross Entropy - One hot encoding

```
softmax_data = [0.7, 0.2, 0.1]
one_hot_data = [1.0, 0.0, 0.0]
softmax = tf.placeholder(tf.float32)
one_hot = tf.placeholder(tf.float32)
cross_entropy = -tf.reduce_sum(tf.multiply
(one_hot, tf.log(softmax)))

with tf.Session() as sess:
    print(sess.run(cross_entropy, feed_dict=
    {softmax: softmax_data, one_hot:
    one_hot_data}))
```

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction to Tensorflow

Tensorflow Mini-batching

- Mini-batching is a technique for training on subsets of the dataset instead of all the data at one time.
- This-provides the ability to train a model, even if a computer lacks the memory to store the entire dataset.
- Mini-batching is computationally inefficient, since you can't calculate the loss simultaneously across all samples.

Introduction to Tensorflow

Epochs

- An epoch is a single forward and backward pass of the whole dataset.
- This is used to increase the accuracy of the model without requiring more data.

Introduction to Tensorflow

Training

- The dataset is divided into training, validation and test sets.
- Learning rate shall be selected appropriately.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

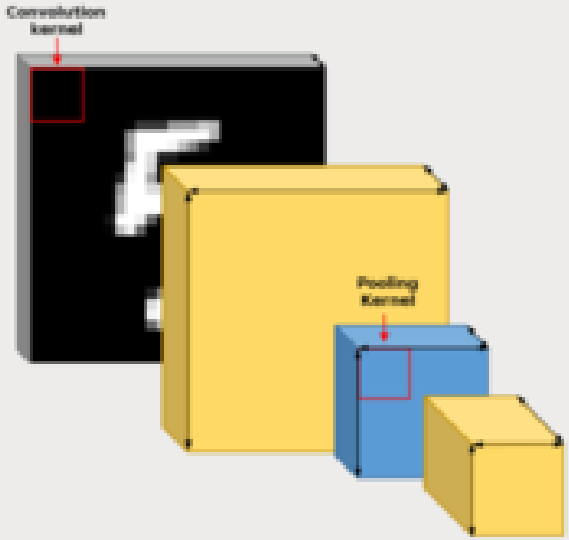
Behavioral cloning

Accurate and efficient vehicle detection

Traffic sign classification

Convolution & Pooling

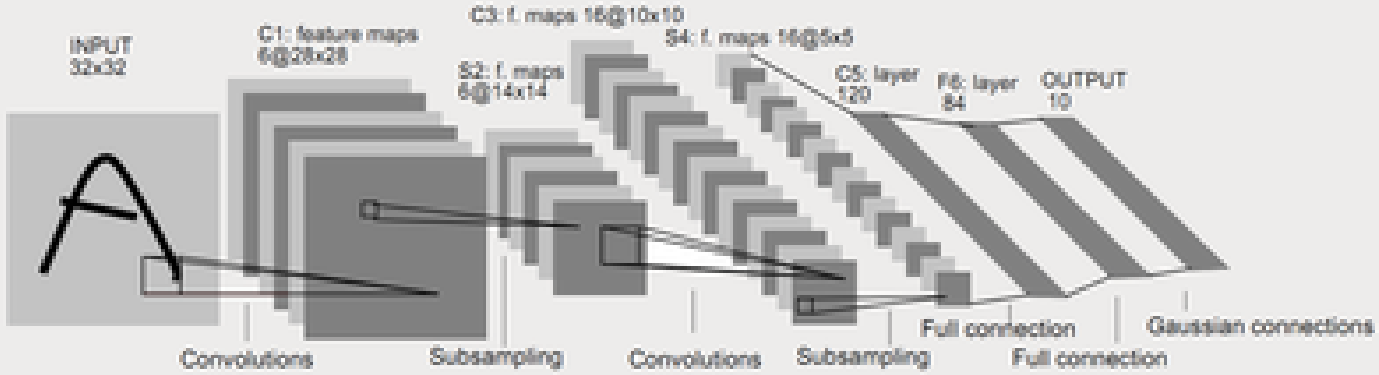
- The first layer in a CNN is always a Convolutional Layer.
- A kernel is convolved with the input layer, producing results for the next layer.
- Subsampling is applied in the form of Max Pool in the LeNet architecture.



Traffic sign classification

LeNet Architecture

- A 32×32 image.
- Convolutional layer.
- Subsampling layer.
- Sequence of convolution layers
- Sequence of pooling layers.
- 3 fully connected layers.



Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

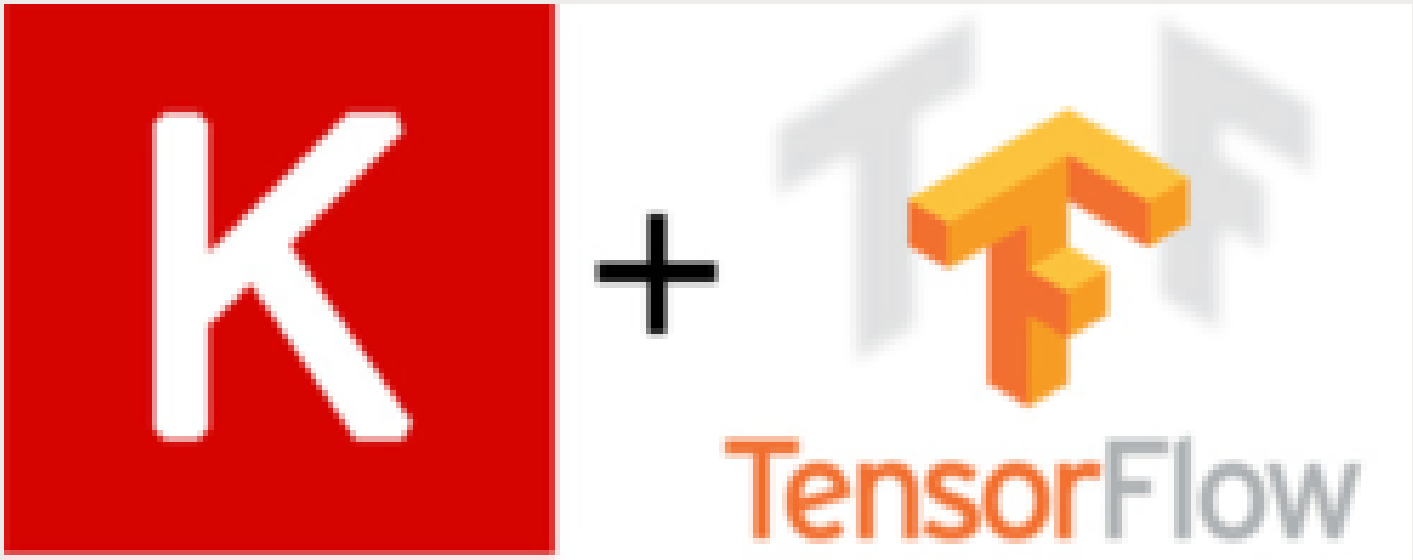
Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection



Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction to Keras

Keras Models

- There are two types of models in Keras:
  - Sequential: The Sequential model is a linear stack of layers.
  - Model (functional API): In the functional API, given some input tensor(s) and output tensor(s), you can instantiate a Model.

Introduction to Keras

Keras Layers

- Keras will automatically infer the shape of all layers after the rst layer. This means you only have to set the input dimensions for the rst layer.
- A Keras layer is just like a neural network layer.
- There are many types of layers in Keras:
  - Core layers.
  - Convolutional layers.
  - Pooling layers
  - Locally connected layers
  - Activation layers.
  - ...

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Behavioral cloning

Car Driving Simulator

- A car draving simulator is used for beharioral cloning.
- 3 cameras are installed on the car, grabbing left, right and center images.
- In the training mode, drive the car like in a video game.
- Collect the training data using the record button.
- Collected images are stored in a local directory.
- The autonomous mode is used after the training of the network.
- The trained network can drive the car inside the road.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Behavioral cloning

Strategies for Collecting Data

- The car should stay in the center of the road as much as possible.
- If the car veers o to the side, it should recover back to center.
- Driving counter-clockwise can help the model generalize.
- flipping the images is a quick way to augment the data.
- Collecting data from the second track can also help generalize the model.
- We want to avoid over tting or under tting when training the model.
- Knowing when to stop collecting more data.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Behavioral cloning

Training a Network

- Run the simulator and select a driving track.
- Press the record button and start driving.
- left, right and center images are stored in a local directory.
- The steering angle for each image is stored in a CSV file.
- Build in a network in Keras.
- Use grabbed images for training of the network.
- Use 80% of the images for training and 20% for validation.
- The training and validation can be 70% and 30%.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Behavioral cloning

Training a Network ...

- In case of poor predictions on both training and validation sets (high mean square on both):
  - Increase the number of epochs.
  - Add more convolutions to the network.
- In case of good predictions of the training set but poor predictions on the validations set (overfitting):
  - Use drop out or pooling layers.
  - Use fewer convolution or fewer connected layers.
  - Collect more data or furthere augment the data set.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Behavioral cloning

Training a Network ...

- Training using un-normalized data results in poor car driving.
- In order to normalize the images, the pixel values are divided by 255.
- In order to mean center the data, each element is subtracted by 0.5.
- Flipping images improves the quality of training.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Behavioral cloning

Training a Network ...

some general guidelines for data collection:

- Two or three laps of center lane driving.
- One lap of recovery driving from the sides.
- One lap focusing on driving smoothly around curves.

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

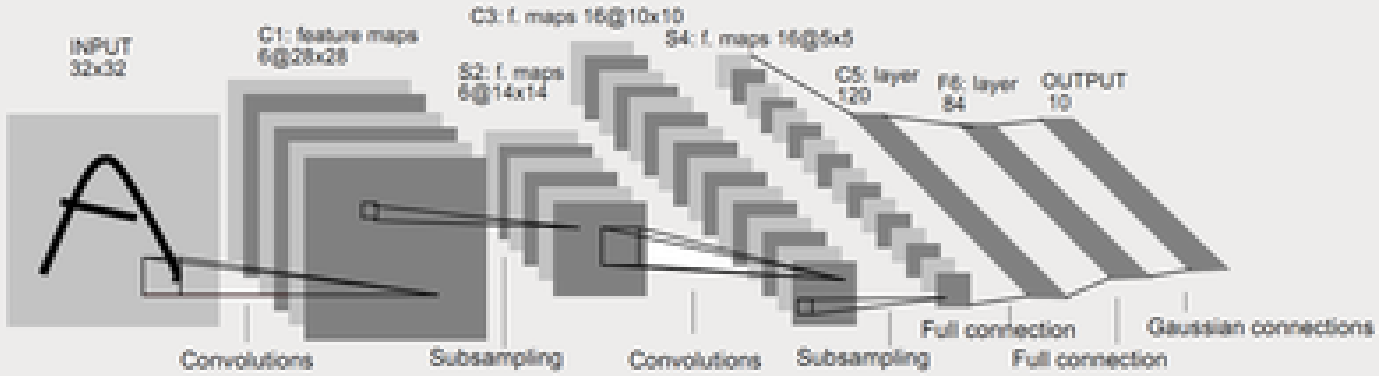
Traffic sign classification by TensorFlow

Introduction to Keras

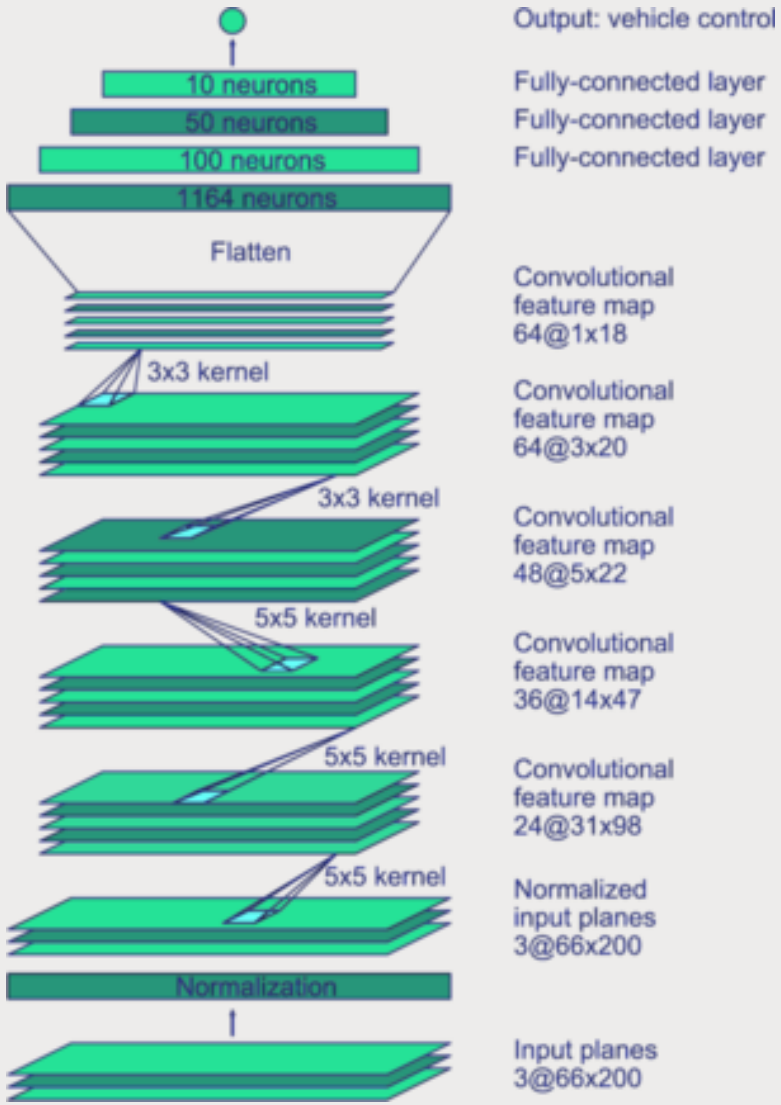
Behavioral cloning

Accurate and efficient vehicle detection

Training by LeNet architecture



Training by Nvidia architecture



Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Behavioral cloning

Defining Nvidia network in Keras

```
from keras.models import Sequential
from keras.layers import Flatten, Dense,
Lambda, Convolution2D, Cropping2D, Dropout
from keras.layers.pooling import MaxPooling2D

model = Sequential()
model.add(Cropping2D(cropping=((50,20),
(0,0)), input_shape=(160,320,3)))
model.add(Lambda(lambda x: x/255. - 0.5))
model.add(Convolution2D(24,5,5,subsample=
(2,2),activation='relu'))
...
model.add(Flatten())
```

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Behavioral cloning

Defining Nvidia network in Keras

```
model.compile(loss='mse',optimizer='adam')

model.fit_generator( train_generator,
    steps_per_epoch = train_steps,
    epochs=5 )
```

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection

Introduction to Deep Neural Networks

Finding lane lines by low-level image processing techniques

Introduction to TensorFlow

Traffic sign classification by TensorFlow

Introduction to Keras

Behavioral cloning

Accurate and efficient vehicle detection