PROBLEM DEFINITION

• The development of effective autonomous vehicles is a popular application of machine learning and control.
• We wish to solve a learning problem predicting steering angles \([a_1, a_m]^T\) over the course of a road segment from Udacity's low-resolution images \([X_1 ... X_m]^T\) where each \(X_i\) is defined by a 640x480x3 RGB tensor.
• We partition the driving data with a 70:30 train and test split.

IMAGE PRE-PROCESSING

• [1] Eliminate top-half of the images, and downsample by a factor of 100. This size captures necessary information for prediction and is of low enough dimensionality to allow training a network to be computationally feasible.
• [2] Apply edge-detection: Construct an image \(X'\) based on thresholding the Sobel kernel operator on input \(X\):

\[
X' = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * X
\]

Define \((S_X, S_Y) = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} * X'\) for each \((i,j)\) in the output image, set \(X_{ij} = 255\) if either of the following are true, and set \(X_{ij} = 0\) otherwise.

• Sobel gradient magnitude \(\sqrt{S_X(i,j)^2 + S_Y(i,j)^2}\) is above some cutoff threshold (preserving only the edges).
• Grayscale value of the pixel \((i,j)\) is above some cutoff threshold (preserving only white or near-white sections of the image). Together, lane capture is reasonable.

CONVOLUTIONAL NEURAL NETWORK MODEL

We discretize the problem: each steering angle gets a class label \(x \in [1,1101]\), where each label represents a range of \(0.01\) rad.

**Forward Propagation:** \(z^{(l+1)} = W^{(l)}a^{(l)}\) and \(a^{(l+1)} = f_l(z^{(l)})\) with \(a^{(1)} = x_i\) and \(h_{W,b} = a^{(l+1)}\). Layer \(l\) has activation func. \(f^{(l)}\).

Cost \(|n\) examples\(|: f(W, b) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{2} \|h_{W,b}(x_i) - y_i\|^2\)

Training uses the Back-Propagation algorithm to optimize the cost \(J\). The update rule is: \(\Delta W^{(l)} := \Delta W^{(l)} + \nabla_W J^{(l)}\) with \(W^{(l)} = W^{(l)} - \alpha \left(\frac{1}{m} \Delta W^{(l)}\right)\).

For Typical Layers:

\[
\delta^{(l+1)} = \delta^{(l+1)}(a^{(l)})^T
\]

For Convolutional Layers

\[
\delta^{(l)} = \sum_{d=1}^{m} \nabla_{W_d} J^{(l)} = \nabla_{W_d} J^{(l)} \delta^{(l+1)} f'(a^{(l)})
\]

RESULTS

- For training and testing, we considered in particular highway driving. The conv. neural network was effective at predicting steering angle within reasonable error.
- Sobel pre-filtering of images provided very marginal benefits (between 0.001 and 0.007 RMSE on test sets).
- RMSE of the above test segment was 0.031 (without Sobel preprocessing) and 0.034 (with preprocessing).

FUTURE WORK

- Introduce more class labels to better approximate angle continuity (for this, more data must be collected to provide sufficient training volume for higher-magnitude labels).
- Introduce additional conv. layers to reduce the bias of the model, and run stochastic gradient descent (on resources with more computational power) for a greater number of epochs for better convergence at local optima.
- Compute cross-image gradients that can determine directions of feature change to better predict turn angles.

REFERENCES

Cameron, Oliver. Open Sourcing 223GB of Driving Data. Medium, 5 Oct. '16.
http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks