Applying Techniques in Supervised Deep Learning to Steering Angle Prediction in Autonomous Vehicles

Vinay Sriram, Petar Penkov, James Ye

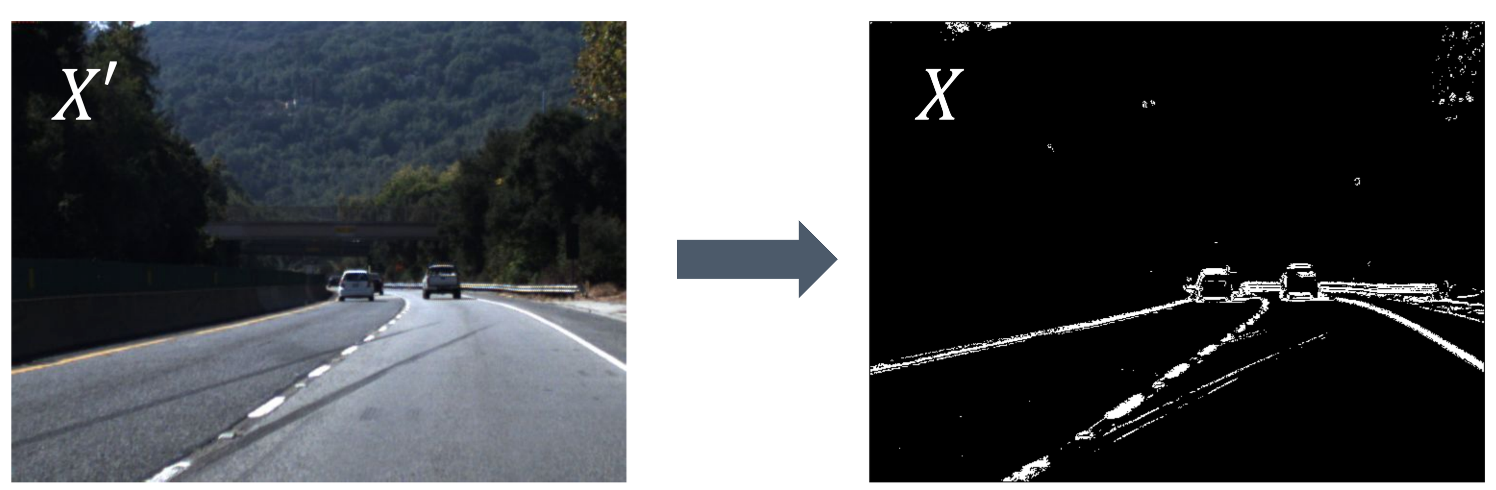
Stanford University

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 $[\alpha_1 \dots \alpha_m]^T$ over the course of a road segment from Udacity's low-resolution images $[X_1 \dots X_m]^T$ where each X_i is defined by a $640 \times 480 \times 3$ RGB tensor.
- We partition the driving data with a 70:30 train-to-test ratio such that the test-set contains significant turning.

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 and the Sobel kernel operator on input X':



Define $(S_X, S_Y) = \left(\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * X', \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * X' \right)$

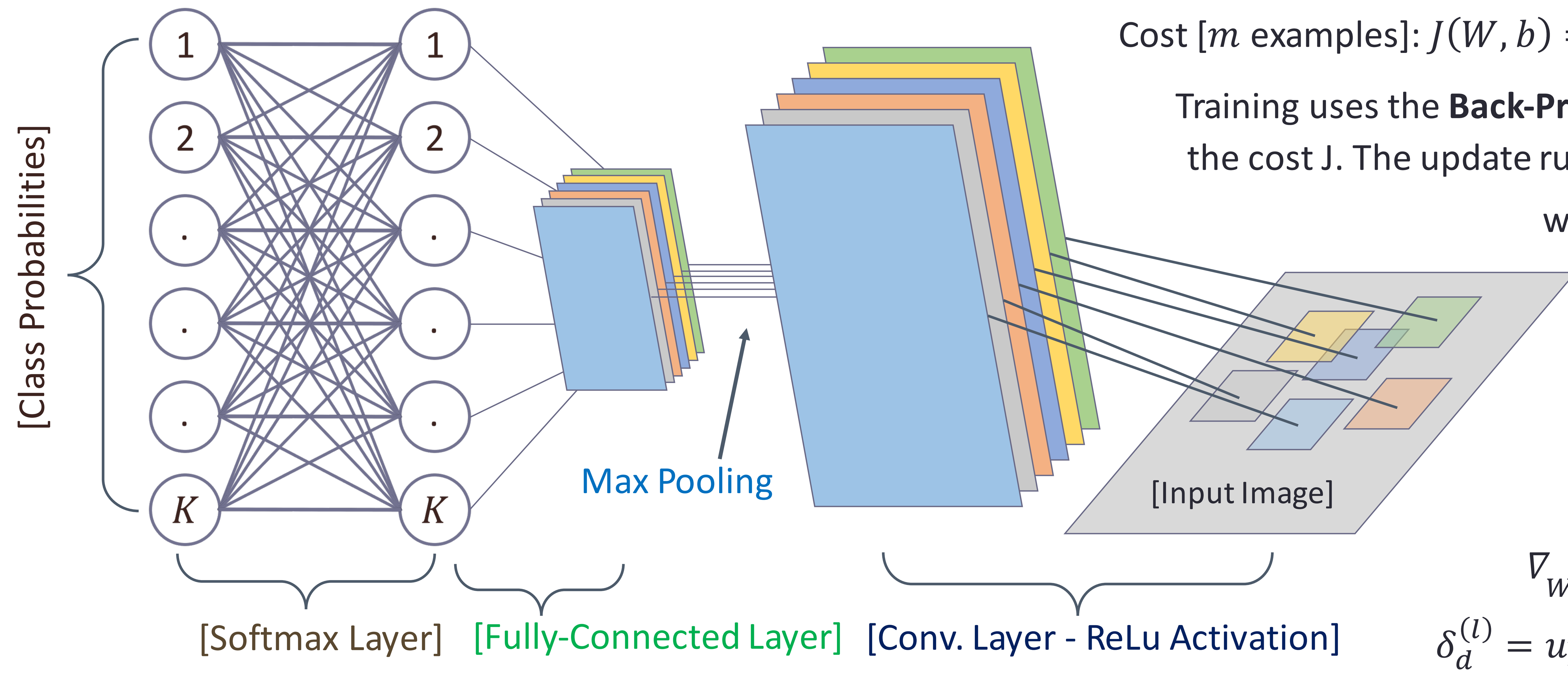
For each (i, j) in the output image, set $X_{ij} = 255$ if either of the following are true, and set $X_{ij} = 0$ otherwise.

- 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 $x \in [1,101]$, 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 $a_i^{(1)} = x_i$ and $h_{W,b} = a^{(L+1)}$. Layer l has activation func. $f^{(l)}$.



Cost [m examples]: $J(W,b) = \left[\frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} \left\| h_{W,b}(x_i) - y_i \right\|^2 \right]$

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

<u>For Typical Layers:</u>

$$\nabla_{W^{(l)}} J = \delta^{(l+1)} \left(a^{(l)}\right)^T$$

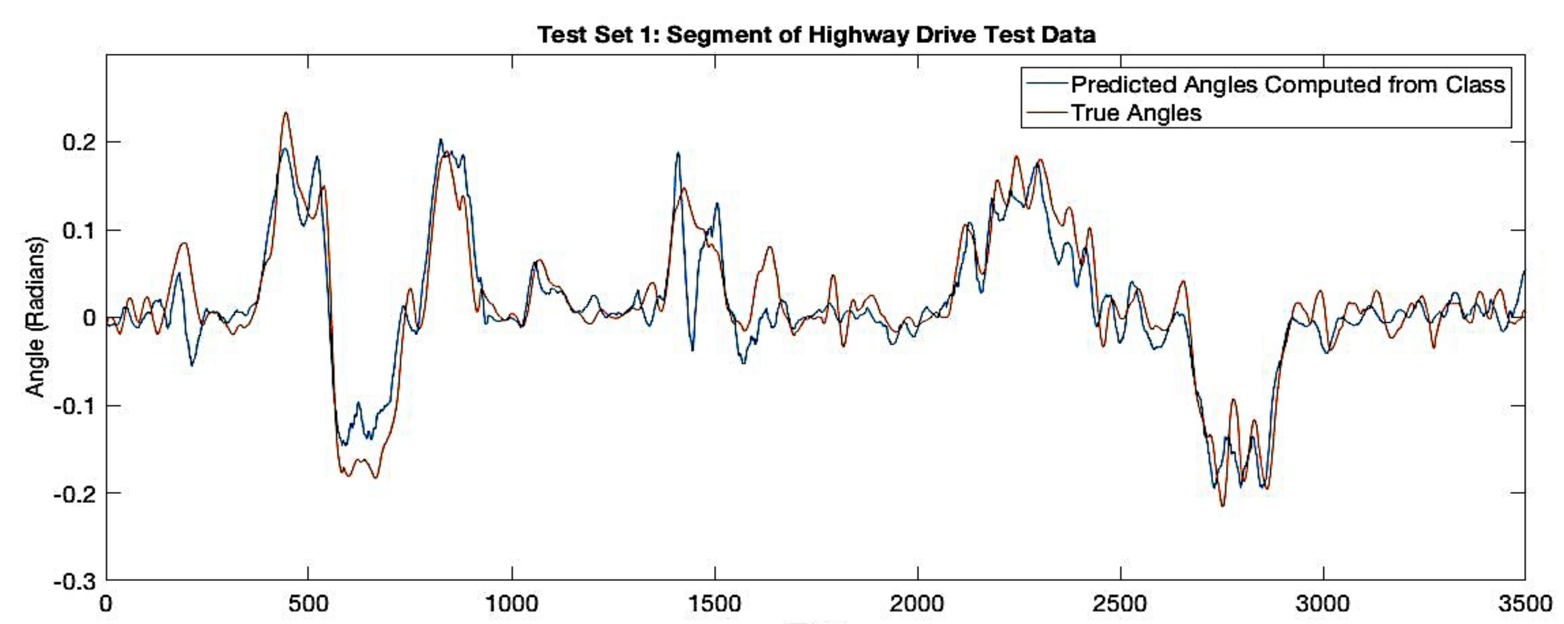
$$\delta^{(l)} = {W^{(l)}}^T \delta^{(l+1)} f'(z^{(l)})$$

<u>For Convolutional Layers</u>

$$\nabla_{W_d^{(l)}} J = \sum_{i=1}^{m} \left(a^{(l)}\right) * rot(\delta_d^{(l+1)})$$

$$\delta_d^{(l)} = upsample({W_d^{(l)}}^T \delta_d^{(l+1)}) f'(z_d^{(l)})$$

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 a 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.

Unsupervised Feature Learning and Deep Learning Tutorial. Stanford. Web. http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks

Pomerleau, Dean. *Neural Net Perception for Mobile Robot Guidance*. (1993)