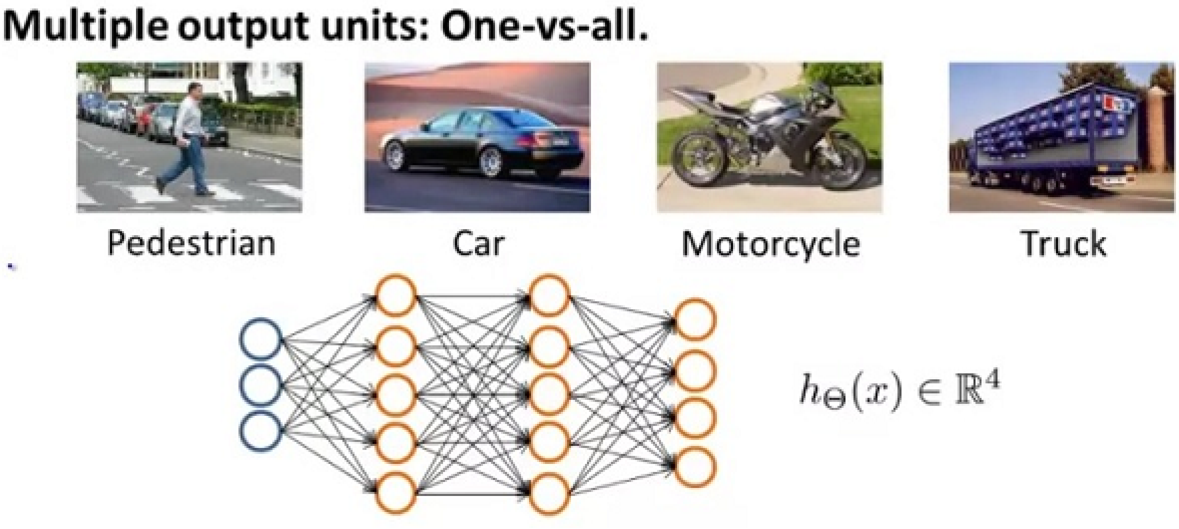
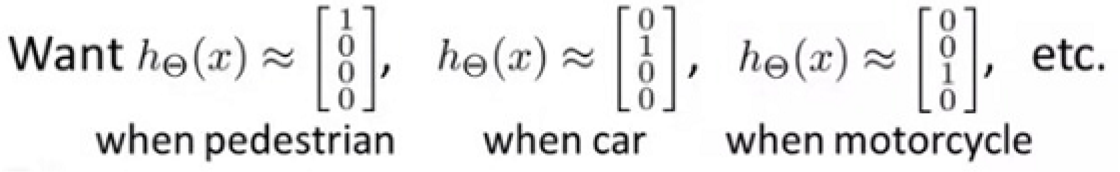
~ Multiclass Classification ~

The way we do multiclass classification in a neural network is essentially an extension of the one versus all method. Let's say that we have a computer vision example and we're trying to recognize four categories of objects and given an image we want to decide if it is a pedestrian, a car, a motorcycle or a truck.



Multiple output units: One-vs-all.

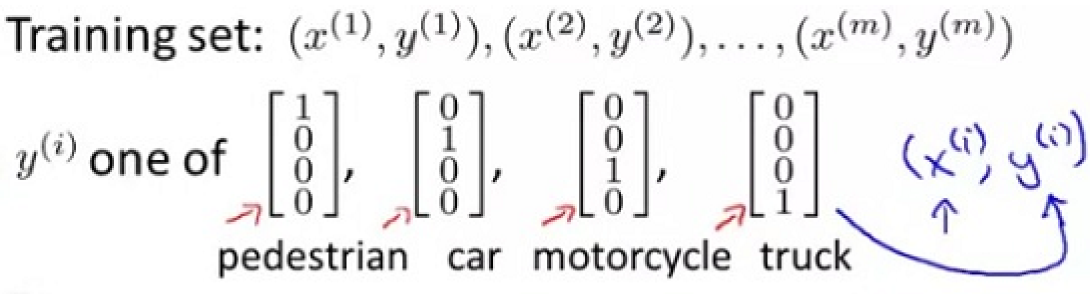
What we would do is build a neural network with four output units so that our neural network now outputs a vector of four numbers. So, the output now is actually needing to be a vector of four numbers and what we're going to try to do is get the first output unit to classify if the first image is a **pedestrian**, the second unit classify if the image is a **car**, the third unit classify if the image is a **motorcycle** and finally the last output unit classify if the fourth image is a **truck**.



Want $h_\Theta(x) \approx \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$, etc.

when pedestrian when car when motorcycle

When image is a pedestrian we get $\begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$ and so on, just like one vs. all described earlier and here we have four logistic regression classifiers. And the way we're going to represent the training set in these settings is as follows.



Training set: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$

$y^{(i)}$ one of $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}$, $\begin{bmatrix}0\\1\\0\\0\end{bmatrix}$, $\begin{bmatrix}0\\0\\1\\0\end{bmatrix}$, $\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$ $(x^{(i)}, y^{(i)})$

pedestrian car motorcycle truck

So, when we have a training set with different images of pedestrians, cars, motorcycles and trucks, what we're going to do in this example is represent the $y^{(i)}$ as a row vector ($y^{(i)} \in \mathbb{R}^{i}$).

Video Question: Suppose you have a multi-class classification problem with 10 classes. Your neural network has 3 layers, and the hidden layer (layer 2) has 5 units. Using the one-vs-all method described here, how many elements does $\Theta^{(2)}$ have?

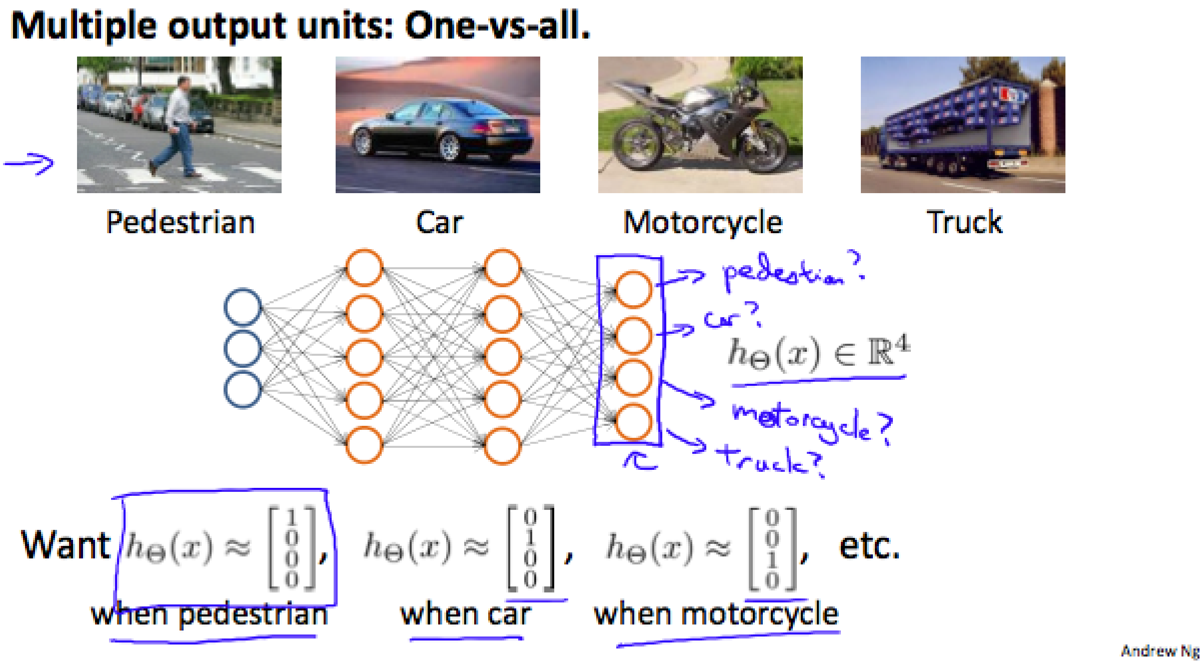
- 50
- 55

60

- 66

Summary

To classify data into multiple classes, we let our hypothesis function return a vector of values. Say we wanted to classify our data into one of four categories. We will use the following example to see how this classification is done. This algorithm takes as input an image and classifies it accordingly:



Multiple output units: One-vs-all.

Want $h_\Theta(x) \approx \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$, $h_\Theta(x) \approx \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$, etc.

when pedestrian when car when motorcycle

Andrew Ng

We can define our set of resulting classes as y:

$$y^{(i)} = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix},$$

Each $y^{(i)}$ represents a different image corresponding to either a car, pedestrian, truck, or motorcycle. The inner layers, each provide us with some new information which leads to our final hypothesis function. The setup looks like:

$$\begin{bmatrix}x_0\\x_1\\x_2\\\cdots\\x_n\end{bmatrix} \rightarrow \begin{bmatrix}a_0^{(2)}\\a_1^{(2)}\\a_2^{(2)}\\\cdots\end{bmatrix} \rightarrow \begin{bmatrix}a_0^{(3)}\\a_1^{(3)}\\a_2^{(3)}\\\cdots\end{bmatrix} \rightarrow \cdots \rightarrow \begin{bmatrix}h_\Theta(x)_1\\h_\Theta(x)_2\\h_\Theta(x)_3\\h_\Theta(x)_4\end{bmatrix}$$

Our resulting hypothesis for one set of inputs may look like:

$$h_\Theta(x) = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$$

In which case our resulting class is the third one down, or $h_\Theta(x)_3$, which represents the motorcycle.