Homework 2

16-311: Introduction to Robotics

Contents

Learning Objectives

1. Gain experience creating and using filters.
2. Practice geometric techniques for finding distance using similar triangles and stereo vision.
3. Use a neural net to classify images.

1 Filters

In this section, we will be discussing filters (also known as masks or kernels). Filtering an image can be useful to average the value of a group of pixels and reduce noise or missing data in the image file.

1. Come up with a 3x3 filter that can take the average of the nine pixels it overlaps with.

2. Come up with a 3x3 filter that takes the average of the neighbors of the center pixel and the center pixel by the 4-point connectivity definition of neighbor.

3. Come up with a slightly more trusting 3x3 filter that gives a weighted average of the 9 pixels that it would overlap with, with more weight on the center pixel. Please make all the weights sum to 1.

4. Come up with a 3x3 filter that could be used to detect vertical lines.

2 Determining Distance Geometrically

Suppose you have a robot with a camera that you need to characterize. This robot has been in the lab for a while so there is little documentation available and the sensing box has been welded shut so you cannot measure the camera's geometry. You want to find the distance between the lens and image sensor (some sources call this the focal length) so you can use that for future experiments to judge distance.

You take a picture of a one-inch square cube from a precise distance of one foot with the cube centered in the image. You threshold and segment the image and count the number of pixels that represent the cube. You repeat this process 10 times and get an average of 47 pixels for the width of the one-inch cube. You read someone's well-documented code for displaying the images from this camera and see that the camera's resolution is 100 ppi (pixels per inch).

Using this information, what is the distance between the lens and image sensor? Please include at least one symbolic equation that you used with your answer.

3 Stereo Vision

The goal of this section is to practice implementing the geometry involved in determining depth from an image and position from stereo cameras. The image below

shows a robot with two forward-facing cameras. The cameras are 10 cm apart and you can simplify the cameras as pinhole cameras with 50 mm between the camera opening and image sensor.

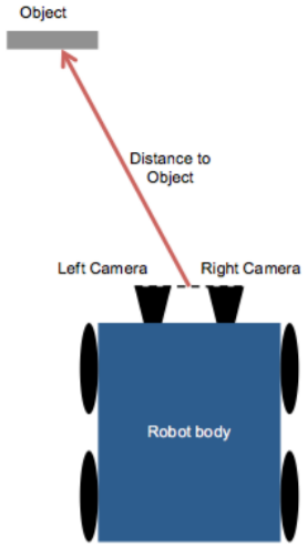

Figure 1: A top view of the robot and object. Not to scale.

Two frames from the camera pair are shown below. The object is 30 pixels left of the center of the image taken by the left camera and 50 pixels to the left of the center of the image taken from the right camera. The resolution of the camera is 20 pixels per cm.

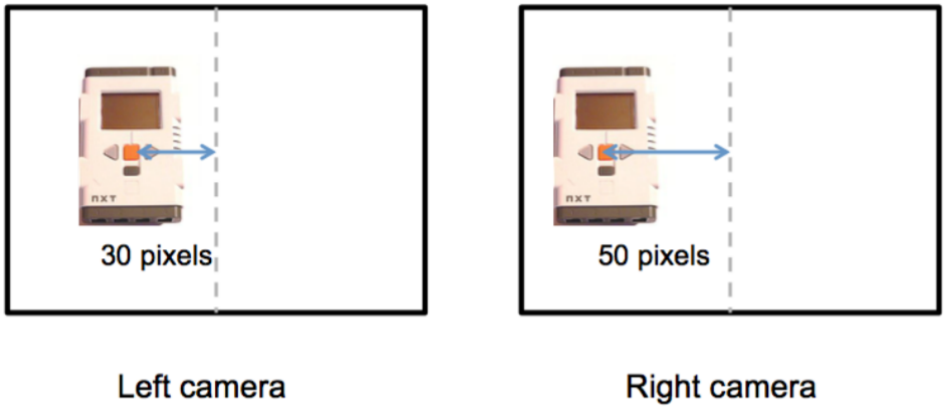

Figure 2: Images from left and right cameras.

In your writeup, find the distance from the center of the front of the lenses to the center of the front of the object (realize that this is not just a single-dimensional problem). Show your work. At a minimum, write down two different symbolic equations showing the geometry used to determine the distance.

4 Neural Net

The explosion of machine leaning has lead to the creation of dozens of machine learning tools. In this question, you will use MATLAB's Deep Learning Designer (pre-installed in MATLAB 2018b). Your image bank will be submitted to Autolab and graded by the TAs using a script (run after the due date). Your responses to the reflection questions will be submitted with the rest of the problems in this assignment and graded by hand on Gradescope.

The goal of this question is to:

- Gain an understanding of the process of creating training data.
- Implement a simple neural net with MATLAB's built in tools.
- Explore the effect of changing some of the neural net parameters for classification.

The purpose of this section is not to get perfect accuracy. This would likely require hundreds of images and multiple layers.

Below are sample images of hedgehogs and porcupines:

Figure 3: Sample images: hedgehog and porcupine.

We are going to classify them as hedgehogs or porcupines. In order to do this, we need to create training and testing (also known as validation) data. Preparing data can be a very time-consuming process in real world problems. Then we will create a neural net. This neural net will train using the portion of our data that we divided as training data and estimate its accuracy using the portion of the data saved for validation. Finally, we will change some parameters of our neural net and look at the effect on performance.

4.1 Creating a Data Set

First, make a folder labeled 'images'. Within this folder, make one folder called 'hedgehogs' and one folder called 'porcupines'. Please follow these titles **exactly** so the TAs can run their code with your images. Populate these folders with images of hedgehogs and porcupines, respectively. Make all the images of **size 200x200x3**. You may want to crop away unneeded parts of the images before resizing. You can use any resize method including imresize. You should only need to make between 15 and 20 images of each category.

To submit this assignment, you will zip up your 'images' folder and submit to Autolab. The TAs will train a NN (described below) on you images, then test on a set of TA test images. The images that the TAs will test your neural net against will be photos from the front of the animal, only. We will use African Pygmy Hedgehogs and North American Porcupines. You will get full credit for your images if your accuracy with our test images is above 50%.

4.2 Creating and Tuning a Neural Net

Next you will create a simple neural net to classify the images. Start from this code and specify the layer characteristics: `http://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/16311/www/current/homework/hw2/p4.m`. This tutorial: MATLAB Deep Learning Classification provides a good overview of the process.

1. Record your accuracy using the parameters given in the starting code. Include a screen shot of the accuracy and loss graphs.
2. Next, note the accuracy with 16 epochs, instead of 8. How does this affect the accuracy? Include a screen shot of the accuracy and loss graphs.
3. Does increasing the number of epochs always help increase accuracy? Explain in one sentence.
4. Choose 10x10 for the filter size. How does this affect accuracy versus the 20x20 filter? Include a screen shot of the accuracy and loss graphs.
5. Finally, try an activation function other than ReLU. How does this affect accuracy? Include a screen shot of the accuracy and loss graphs.

5 Backpropagation

In this section we will ask you to symbolically and numerically solve some backpropagation for the fully connected neural net in figure 4 explained in the following section.

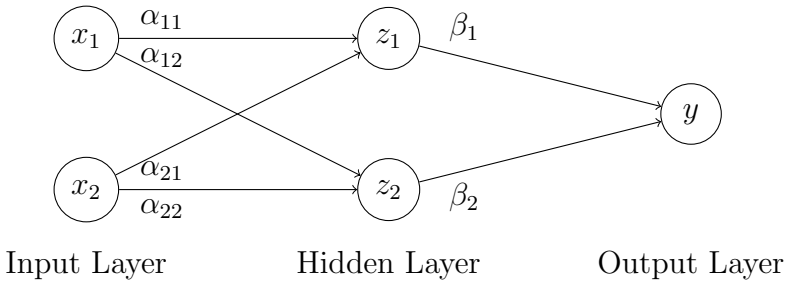

Figure 4: Neural Network

5.1 Forward Pass

The input layer of this Neural Network has two inputs, x_1 and x_2. There is one hidden layer with two nodes, z_1 and z_2. Each of these hidden nodes has a linear activation function applied to it. These are then weighted and fed into the final output layer, y, which uses the sigmoid activation function. This gives us the following equations.

Linear combinations that are inputs into each hidden layer:

$$a_1 = \alpha_{11}x_1 + \alpha_{21}x_2 \tag{1}$$
$$a_2 = \alpha_{12}x_1 + \alpha_{22}x_2 \tag{2}$$

The output of each hidden layer:

$$z_1 = 1.5a_1 \tag{3}$$
$$z_2 = 2.5a_2 \tag{4}$$

Linear combination that are input into the output's activation layer:

$$b = \beta_1 z_1 + \beta_2 z_2 \tag{5}$$

The activation function of the output layer:

$$y = \sigma(b) = \frac{1}{1 + e^{-b}} \tag{6}$$

To calculate the error of the output layer, y, with respect to the target, $\hat{y}$, we will use the squared error function, giving us the equation for loss:

$$\ell = \frac{1}{2}(\hat{y} - y)^2 \tag{7}$$

Assuming the following variables are given these values:

$$x_1 = 2 \qquad \beta_1 = 0.1$$
$$x_2 = 4 \qquad \beta_2 = 0.057$$
$$\alpha = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.3 \\ 0.7 & 0.6 \end{bmatrix} \qquad \hat{y} = 1$$

We calculate the forward pass to be:

$$\begin{aligned} a_1 &= 3.8 & b &= 0.998 \\ a_2 &= 3 & y &= 0.731 \\ z_1 &= 5.7 & \ell &= 0.036 \\ z_2 &= 7.5 \end{aligned}$$

5.2 Backpropagation

Now that we have the forward pass and know our current loss we want to minimize this loss, thus bringing us closer to our target value $\hat{y}$. We can use backpropagation to determine how the changes in the weights of the neural net affect our loss. In order to do this we want to calculate the change in loss with respect to the change in our weights, in other words, the gradient of the loss function.

For this homework we will only ask you to calculate key partial gradients, where in a real neural net you would examine all the gradients. First calculate the partial derivatives for each listed weight symbolically using the chain rule. Your answers for each of the following questions should be partial derivatives with respect to ℓ, a_i, z_i, b, y, α_{ij}, or β_i.

1. $\frac{\partial \ell}{\partial \beta_1}$
2. $\frac{\partial \ell}{\partial \alpha_{11}}$
3. $\frac{\partial \ell}{\partial \alpha_{12}}$
4. $\frac{\partial \ell}{\partial \alpha_{21}}$

Now using the symbolic equations and the values from the forward pass, calculate each of the partial derivatives numerically. You can use the following value for $\frac{\partial \ell}{\partial b}$.

$$\begin{aligned} \frac{\partial \ell}{\partial b} &= (y - \hat{y}) \frac{e^{-b}}{(1 + e^{-b})^2} \\ &= -0.053 \end{aligned}$$

5. $\frac{\partial \ell}{\partial \beta_1}$
6. $\frac{\partial \ell}{\partial \alpha_{11}}$

7. $\frac{\partial \ell}{\partial \alpha_{12}}$
8. $\frac{\partial \ell}{\partial \alpha_{21}}$
9. Conceptually, what does the ratio of $\frac{\partial \ell}{\partial \alpha_{11}}$ to $\frac{\partial \ell}{\partial \alpha_{21}}$ mean with regards to tuning the weights of the neural net and changing the loss.
10. Which weight when changed will have the greatest affect on the loss of the neural net (of the partial derivatives you calculated) and why?

What To Submit

Submissions are due on Autolab/Gradescope by the date specified in the Syllabus.

1. Create a .pdf file with the written answers ALL THE SECTIONS named hw2.pdf and submit to Gradescope.
2. Zip your images folder from Section 4 and submit to Autolab.