Machine Learning for Applications in Computer Vision: Week 2

Liu, Nan
Wiedemann, Andreas

Github: https://github.com/DJLoco/TUMMachineLearning.git

Exercise 2:

Backpropagation with single modified neuron:

- Input x: Set the corresponding activation a_j^1 for the input layer.

- Feedforward: For each $l = 2, 3, \ldots, L$ compute

$$z_j^l = f\left(\sum_j w_j x_j + b^l\right) \text{ and } a_j^l = \sigma(z_j^l)$$

- Output error δ_j^L: ompute the vector

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L) z_j^{l'}(f_j)$$

- Backpropagate the error: For each $l = L-1, L-2, \cdots, 2$ compute

$$\delta_j^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) z^{l'}(f)$$

- Output: The gradient of the cost function is given by

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l \text{ and } \frac{\partial C}{\partial b_j^l} = \delta_j^l$$

Exercise 3.2:

The objective is to get to know the Caffe , a powerful framework of Neural Network and Deep Learning. First of all, to install, compile Caffe and run

the example. Then use the Imagenet network set up in the previous problem to classify each image downloaded from `https://vision.in.tum.de/teaching/ss2015/mlpractice_ss2015/slides`.

Before Classification, we preprocessed the images by setting the input channel order for BRG as needed for the reference ImageNet model, setting the mean to subtract for entering the data, letting the reference model operate on images in [0,255] instead of [0,1] and swapping the channel which makes the reference model has channel in BGR order instead of RGB.

For the 8 images, we use the ImageNet network to predict the class of which, and provide the class name, the probability of that class, the entropy of the prediction. From the class names, we can see that the results are pretty accurate except for image 5 which is classified as a mouse by mistake. This mistake is caused by the lack of enough preprocessing, we suppose. From the definition of the prediction probability (softmax), we can know that the higher the probability, the more accurate the prediction result is. In other words, the lower the probability, the larger the uncertainty is. More than that, we also use the prediction probability to compute the entropy, the larger the entropy, the larger the uncertainty will be. From the eight results, the larger the probability, the smaller the entropy, meaning the larger the certainty, the larger the prediction accuracy.

Reference

`http://neuralnetworksanddeeplearning.com/chap1.html`
`http://neuralnetworksanddeeplearning.com/chap2.html`
`http://caffe.berkeleyvision.org`
`http://caffe.berkeleyvision.org/gathered/examples/imagenet.html`
`http://nbviewer.ipython.org/github/BVLC/caffe/blob/master/examples/filter_visualization.ipynb`
`http://nbviewer.ipython.org/github/BVLC/caffe/blob/master/examples/classification.ipynb`