Machine Learning for Applications in Computer Vision: Week 2

Exercice 2:

Backpropagation with single modified neuron:

- Input x: Set the corresponding activation a_j for the input layer.
- Feed forward: For each $l = 2, 3, \cdots, L$ compute

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

- Output error δ_j^L: ompute the vector

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

- Back propagate 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

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

Exercice 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, and the softmax and entropy