Chris Piech
CS109

Section #9
Dec 5, 2018

Section #9: Deep Learning Soln

---

1. **Run a Tensor Flow Algorithm**

   See the Python notebook solutions in section or online!

2. **Deep Dream**

   a. Loop over all images, run them through your deep learning network, and select the one for which the activation of the final neuron is the largest.

   b. Note that: $h_2 = \sigma(\sum_{i=0}^{64} \theta_{i,2}^{(h)} \mathbf{x}_i)$. We are going to use gradient ascent to choose pixel values ($\mathbf{x}$) that maximize the activation of the neuron (h_2)!

   $$\arg\max_{\mathbf{x}} h_2 = \arg\max_{\mathbf{x}} \sigma\Big(\sum_{i=0}^{64} \theta_{i,2}\mathbf{x}_i\Big)$$

   Which requires us to solve for the derivative of h_2 with respect to each pixel $\mathbf{x}_i$.

   $$\begin{aligned}
   \frac{\partial h_2}{\partial x_i} &= \frac{\partial \sigma(z)}{\partial z} \cdot \frac{\partial z}{\partial x_i} && \text{chain rule where } z = \sum_{k=0}^{64} \theta_{k,2}\mathbf{x}_k \\
   &= \sigma(z)[1-\sigma(z)] \cdot \frac{\partial z}{\partial x_i} && \text{the derivative of the sigmoid function} \\
   &= h_2[1-h_2] \cdot \frac{\partial z}{\partial x_i} && \text{since } h_2 = \sigma(z) \\
   &= h_2[1-h_2] \cdot \theta_i && \text{That's all folks!}
   \end{aligned}$$

   At this point, we have all the tools we need to perform gradient ascent to find the best pixel values to maximize the activation of the h_2 neuron.

   c. The approach is basically the same as in part (b), but we start with the pixels $\mathbf{x}$ set to the input picture's pixel values. We then optimize for the output of $Y = 1$, the cat neuron (using the same methodology of derivatives as above, but with respect to the final neuron instead of h_2). Finally, we only run a few iterations of gradient descent (so that the image is only slightly more catlike, as requested).