Assignment 3 Solution

1. Neural Networks

a.

- Template matching interpretation:
  - rows of θ^T are templates
  - with k rows of θ^T, we have k templates
  - $\theta^T x$ measures how well x matches each of the k templates
  - high similarity to a template of a particular class indicates high membership in this class
- Decision boundary interpretation:
  - rows of θ^T are parameters of k linear discriminant functions
  - each linear discriminant separate one class from all others
  - the value of a linear discriminant is positive for examples belonging to the class and negative otherwise

b.

- For two class classification, we use sigmoid function:
  $\hat{y}_j^{(i)} \equiv P(y=j|x^{(i)}) = \mathrm{sigmoid}(S_j^{(i)}) = \frac{1}{1+\exp(-S_j^{(i)})}$
- For K class classification, we use softmax function:
  $\hat{y}_j^{(i)} \equiv P(y=j|x^{(i)}) = \frac{\exp(S_j^{(i)})}{\sum_{l=1}^{k}\exp(S_l^{(i)})}$

c.

- L1 loss: $L_i = \sum_{j=1}^{k} |\hat{y}_j^{(i)} - y_j^{(i)}|$
- L2 loss: $L_i = \sum_{j=1}^{k} (\hat{y}_j^{(i)} - y_j^{(i)})^2$
- Huber loss:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y-f(x))^2 & \text{if } |y-f(x)| \le \delta, \\ \delta|y-f(x)| - \frac{1}{2}\delta^2 & \text{otherwise} \end{cases}$$

- Cross entropy loss: $L_i = -\sum_{j=1}^{k} y_j^{(i)} \log(\hat{y}_j^{(i)})$

d. The purpose is to get a simpler solution with lower θ which leads to be more stable and generalize better.

e. Opposite direction of the gradient is the direction of greatest decrease

f.

- Gradient descent: update gradient with entire training dataset
- Stochastic gradient descent: update gradient with batch of training dataset

g. Select a initial learning rate (e.g., $10^{-1} \sim 10^{-3}$) and make it smaller as iterations progress

h. To address the poor conditioning, minimum/saddle points, and noisy gradients

i.

- Forward pass: push input to compute all intermediate node values
- Backward pass: start with end nodes push gradients towards the beginning nodes and update the weight
- gradient is propagated in the backward pass

j.

- Fully connected layer: each unit is connected to every unit in subsequent layer which has a large amount of parameters.
- Convolution layer: preserve local spatial neighborhood, convolve input data with filter using stride, multiple filters, and weight sharing within the same layer

k. Dropout is a regularization technique to prevent overfitting. During each training iteration, remove units in fully connected layers with probability of $1-p$. The removed nodes are reinstated with original weights in the subsequent stage. During testing/inference, we do not dropout any nodes.

---

2. Convolution Neural Networks

a.

$$\mathrm{R:}\begin{bmatrix} 9 & 9 \\ 9 & 9 \end{bmatrix} \quad \mathrm{G:}\begin{bmatrix} 18 & 18 \\ 18 & 18 \end{bmatrix} \quad \mathrm{B:}\begin{bmatrix} 18 & 18 \\ 27 & 27 \end{bmatrix} \quad \Rightarrow \mathrm{Final:}\begin{bmatrix} 45 & 45 \\ 54 & 54 \end{bmatrix}$$

b.

$$\mathrm{R:}\begin{bmatrix} 4 & 6 & 6 & 4 \\ 6 & 9 & 9 & 6 \\ 6 & 9 & 9 & 6 \\ 4 & 6 & 6 & 4 \end{bmatrix} \quad \mathrm{G:}\begin{bmatrix} 8 & 12 & 12 & 8 \\ 12 & 18 & 18 & 12 \\ 12 & 18 & 18 & 12 \\ 8 & 12 & 12 & 8 \end{bmatrix} \quad \mathrm{B:}\begin{bmatrix} 6 & 9 & 9 & 6 \\ 12 & 18 & 18 & 12 \\ 18 & 27 & 27 & 18 \\ 14 & 21 & 21 & 14 \end{bmatrix}$$

$$\Rightarrow \mathrm{Final:}\begin{bmatrix} 18 & 27 & 27 & 18 \\ 30 & 45 & 45 & 30 \\ 36 & 54 & 54 & 36 \\ 26 & 39 & 39 & 26 \end{bmatrix}$$

c.

$$\mathrm{R:}\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix} \quad \mathrm{G:}\begin{bmatrix} 8 & 8 \\ 8 & 8 \end{bmatrix} \quad \mathrm{B:}\begin{bmatrix} 12 & 12 \\ 8 & 8 \end{bmatrix} \quad \Rightarrow \mathrm{Final:}\begin{bmatrix} 24 & 24 \\ 20 & 20 \end{bmatrix}$$

d. When we do the convolution, we do the dot product between the filter and image. When filter resemble with the image, we expect a high response. The network is trying to find matches in the image.

e. When pooling between layers (or when using convolution with a stride greater than 1) the spatial dimensions are sampled and so we get an image pyramid with different spatial resolution at the different layers. In this way a fixed size convolution filter covers a larger spatial region in upper layers.

f.

- Increase the number of filters layers by layers
- The purpose is to help in learning more levels of global abstract structures and shrinking the feature space for input to the dense (fully connected) networks.

g. $126 \times 126 \times 16$

h. $\lfloor (W - F + 2P)/S + 1 \rfloor \rightarrow 63 \times 63 \times 16$

i. It can be reduced by the fewer number of filters

j.

- Early convolution layers: extract simple pattern such as edge
- Deeper convolution layers: extract complex pattern

k. Downsampling the spatial dimension

l.

$$R : \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \qquad G : \begin{bmatrix} 2 & 2 \\ 2 & 2 \end{bmatrix} \qquad B : \begin{bmatrix} 2 & 2 \\ 4 & 4 \end{bmatrix}$$