CS231N Assignment 1 - KNN

1 Intuition

1.1 Training and Prediction

- In training stage, KNN simply memorizes the training data (effectively a NOP operation)
- In prediction stage, for each **test** image KNN finds the *nearest* training data (that it previously memorized)
- As number of training samples approaches infinity, KNN can represent any function (subject to many technical conditions)
  - However, this suffers from *curse of dimensionality*
  - For uniform coverage of training space, number of training points increases exponentially with dimension

1.2 Distance Metrics

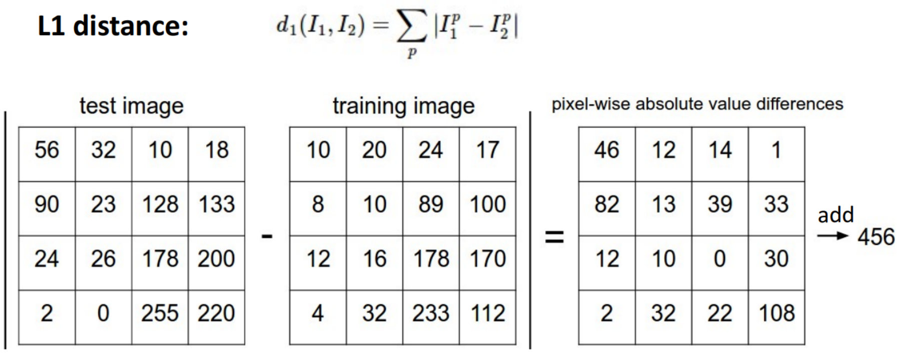

Figure 1

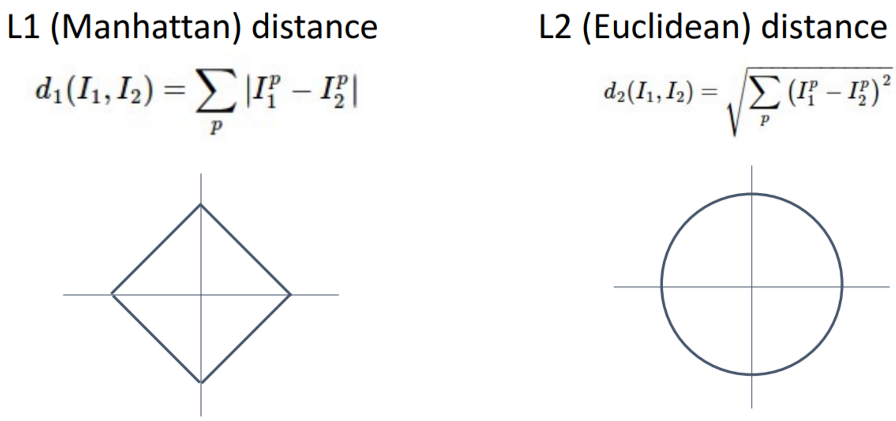

Figure 2

2 Per-pixel calculations

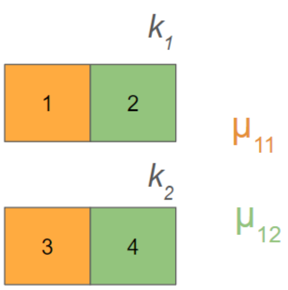

Figure 3: $p^{(k_1)}$ and $p^{(k_2)}$

- General mean $\mu = \frac{1+2+3+4}{1*2*2}$
- Pixel-wise mean $\mu_{ij} = \begin{bmatrix} \mu_{11} \\ \mu_{12} \end{bmatrix} = \begin{bmatrix} \frac{1+3}{2} \\ \frac{2+4}{2} \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$

3 L2 Distance - Fully vectorised

$$
\begin{aligned}
d_2(A,B) &= \sqrt{\sum_p (A_p - B_p)^2} \text{ ; Where } \boldsymbol{A} \text{ and } \boldsymbol{B} \text{ are vectors indexed via } p \\
&= \sqrt{\sum_p \Big\{(A_p)^2 - 2(A_p \cdot B_p) + (B_p)^2\Big\}} \\
&= \sqrt{\Big\{\sum_p (A_p)^2\Big\} - 2\Big\{\sum_p (A_p \cdot B_p)\Big\} + \Big\{\sum_p (B_p)^2\Big\}} \\
&= \sqrt{\Big\{\sum_p (A_p)^2\Big\} - 2\Big\{\boldsymbol{AB}^T\Big\} + \Big\{\sum_p (B_p)^2\Big\}} \\
&= \sqrt{\Big\{\sum_p (A_p)^2\Big\} + \Big\{\sum_p (B_p)^2\Big\} - 2\Big\{\boldsymbol{AB}^T\Big\}}
\end{aligned}
$$

3.1 Context

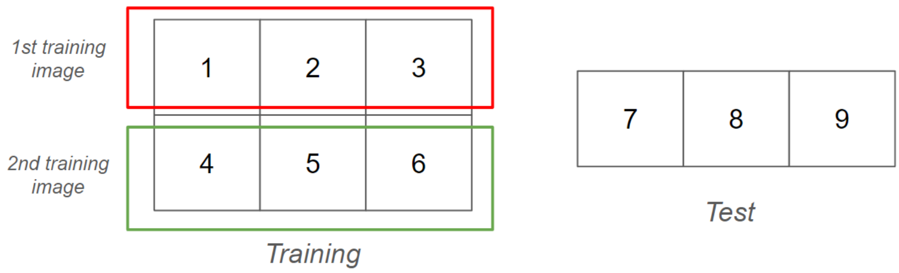

Figure 4

- Training $= \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$, Test $= \begin{bmatrix} 7 & 8 & 9 \end{bmatrix}$
- We want to calculate the L2 distance between test image and every training image

3.1.1 One loop

1. Broadcast; $(Tr - Te)_{broadcast} = \begin{bmatrix} 1-7 & 2-8 & 3-9 \\ 4-7 & 5-8 & 6-9 \end{bmatrix} = \begin{bmatrix} -6 & -6 & -6 \\ -3 & -3 & -3 \end{bmatrix}$
2. Square terms; $(Tr - Te)^2_{broadcast} = \begin{bmatrix} 36 & 36 & 36 \\ 9 & 9 & 9 \end{bmatrix}$
3. Sum along rows; $\sum_{rows} (Tr - Te)^2_{broadcast} = \begin{bmatrix} 36+36+36 \\ 9+9+9 \end{bmatrix} = \begin{bmatrix} 108 \\ 27 \end{bmatrix}$
4. Square root; $\sqrt{\sum_{rows} (Tr - Te)^2_{broadcast}} = \begin{bmatrix} \sqrt{108} \\ \sqrt{27} \end{bmatrix}$

3.1.2 No loop

1. Square **elements of** training matrix; $Tr_{squared} = \begin{bmatrix} 1 & 4 & 9 \\ 16 & 25 & 36 \end{bmatrix}$

   1.1. $\sum_{row} Tr_{squared} = \begin{bmatrix} 1+4+9 \\ 16+25+36 \end{bmatrix} = \begin{bmatrix} 14 \\ 77 \end{bmatrix}$

2. Square **elements of** test matrix; $Te_{squared} = \begin{bmatrix} 49 & 64 & 81 \end{bmatrix}$

   2.1. $\sum_{row} Te_{squared} = \begin{bmatrix} 49+64+81 \end{bmatrix} = \begin{bmatrix} 194 \end{bmatrix}$

3. Broadcast; $\Big(\sum_{row} Tr_{squared} + \sum_{row} Te_{squared}\Big)_{broadcast} = \begin{bmatrix} 14+194 \\ 77+194 \end{bmatrix} = \begin{bmatrix} 208 \\ 271 \end{bmatrix}$
4. $2(\boldsymbol{TrTe}^T) = 2 \cdot \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} = \begin{bmatrix} 100 \\ 244 \end{bmatrix}$

5. $\left(\sum_{row} Tr_{squared} + \sum_{row} Te_{squared}\right)_{broadcast} - 2(\boldsymbol{TrTe}^T)$