CMPE258 PART II

Oct. 20, 22

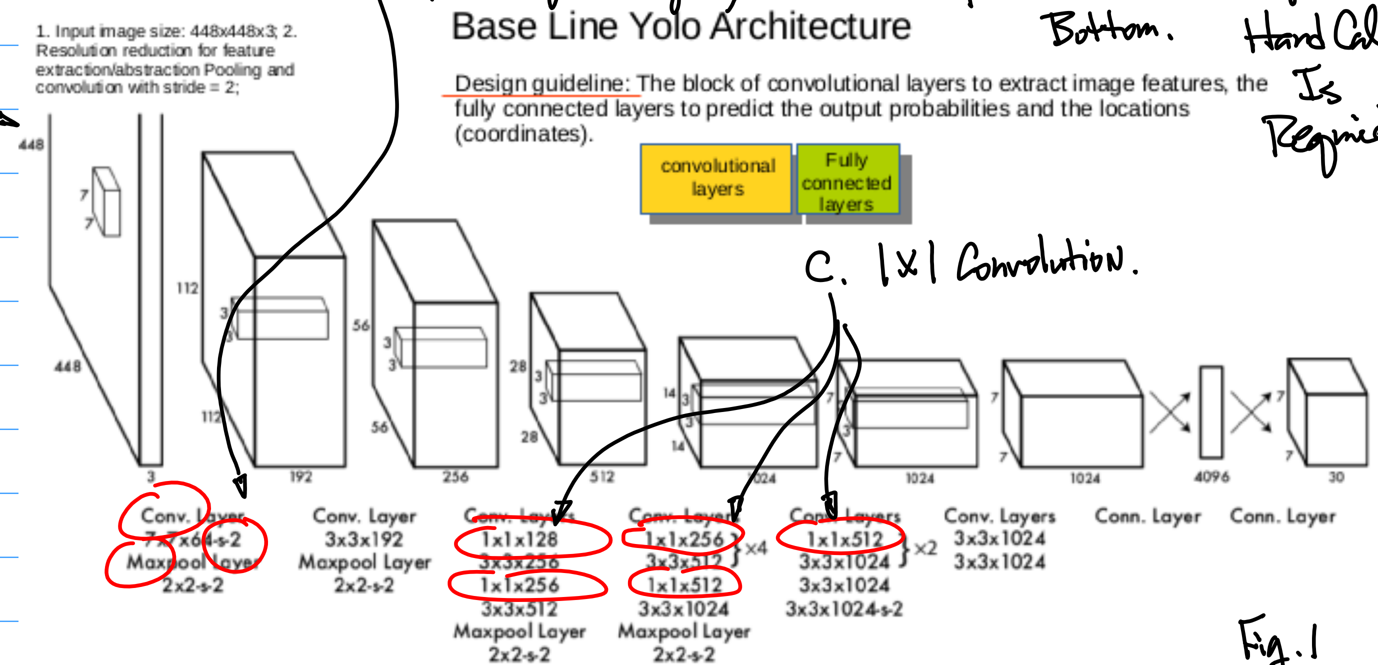
40.

a. 448x448x3

Channel

b. Convolution with Stride = 2. "Scan" the input Layer at every two pixels from Left to Right, then at every other Row from Top to Bottom.

Hand Calculation Is Required.



c. 1x1 Convolution.

Fig. 1

The Motivation for the 2-Stride, or K-stride convolution is to combine Convolution with Subsampling, e.g. Reduction of the feature layer. Example Below, See Stride pattern in Green.

Consider 3x3 Kernel convolution

20	20	0	0
20	20	0	0
10	10	0	0
10	10	0	0

Fig. 3a

1	0	-1
1	0	-1
1	0	-1

3x3x1 Channel

Image

20	20	0	0
20	20	0	0
10	10	0	0
10	10	0	0

Fig. 2

Reduce 3x3 kernel to 1x1 Kernel below, then Convole with the image $I(x,y) \longrightarrow$ Output Resolution is MxN

MxN

$\boxed{2.5}$

X-Y Two Dimensional Space "Spatial"

V.S. Temporal (Time)

CmpE258

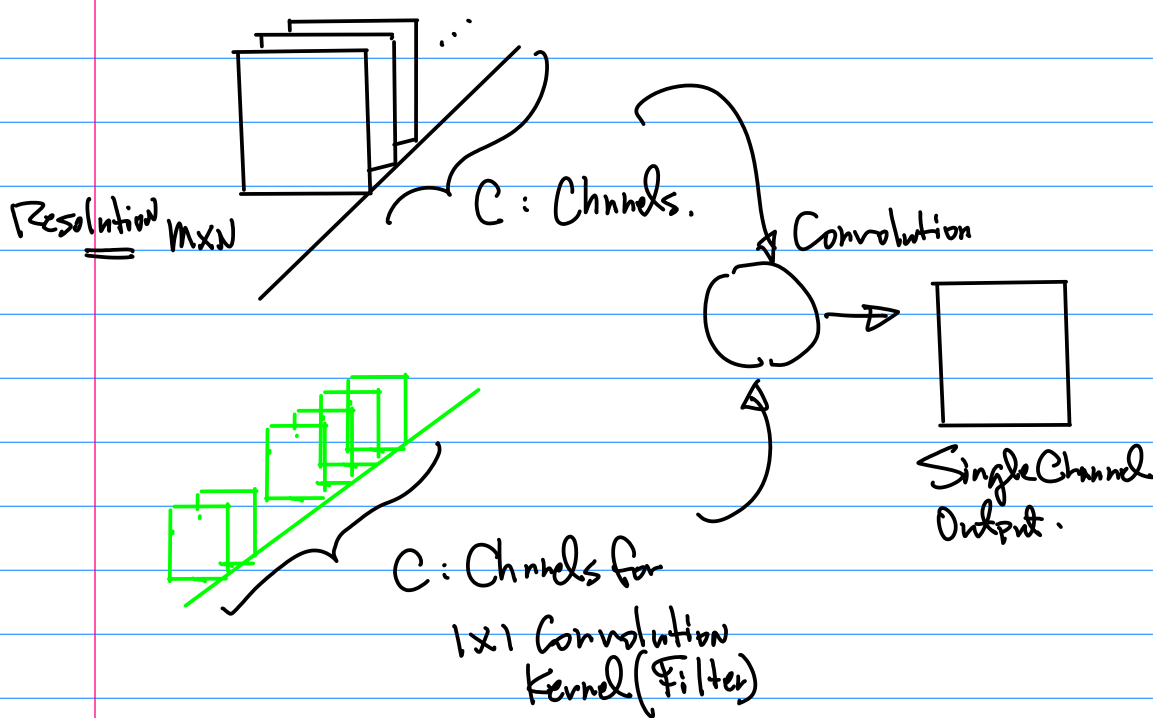
Oct. 20, 22

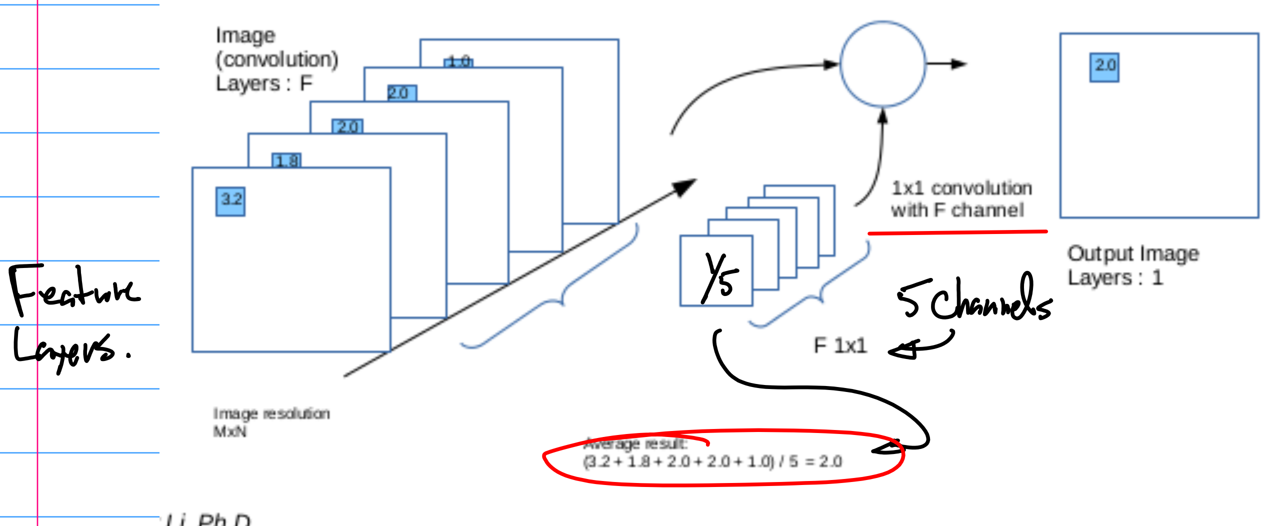
Space (Spatial) in 2D → Space in 3D and Beyond.

Note: 1° 1x1 Convolution Reduces the Number of Layers of the Input.

2° Pooling.

for Example for Image





CmpE258

Oct. 20, 22

Example: a) A loss function can be defined by subtracting the Output (function) from the ground truth. → Square it to prevent from possible Cancellations when Summed up together.

b) One Loss function

$$f_{Loss_\Sigma} = \sum_{i \in I} f_{Loss,i} \quad \ldots (1)$$

$$f_{Loss_\Sigma} = \alpha_1 f_{Loss_1} + \alpha_2 f_{Loss_2} + \cdots + \alpha_K f_{Loss_K}$$

where $\alpha_1 + \alpha_2 + \cdots + \alpha_K = 1$

Location Based Loss function. c)

Loss Function for YOLO

Location $(\hat{x}_i, \hat{y}_i)$ of Bounding Box

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2-1} \sum_{j=0}^{B-1} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \lambda_{\textbf{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

$\mathbb{1}_{ij}$: i for Each Grid S^2; j for Bounding Boxes $B+1$

$$\mathbb{1}_{ij} = \begin{cases} 1 & \text{when } B_{ij} \text{ exists} \\ 0 & \text{o/w (otherwise)} \end{cases}$$

Note: If we want emphasize Location Prediction, → f_{Loss_1} → $\alpha_1 \uparrow$ Bigger α_1

d) Geometric Shape (Always in Rectangle Shape, Size) Loss Function

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right]$$

c. From Eqn Below.

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \underbrace{\Pr(\text{Class}_i)}_{\text{Lower Branch}} * \underbrace{\text{IOU}_{\text{pred}}^{\text{truth}}}_{\text{Upper Branch}} \quad (1)$$

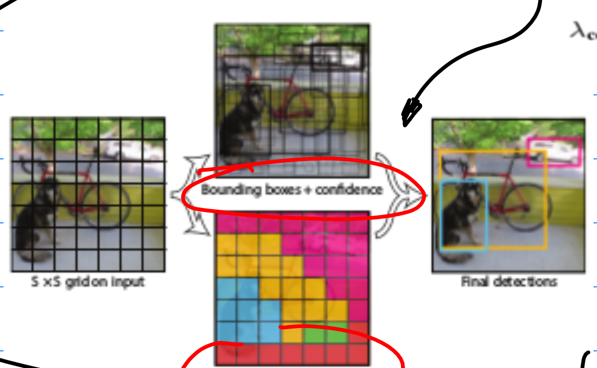
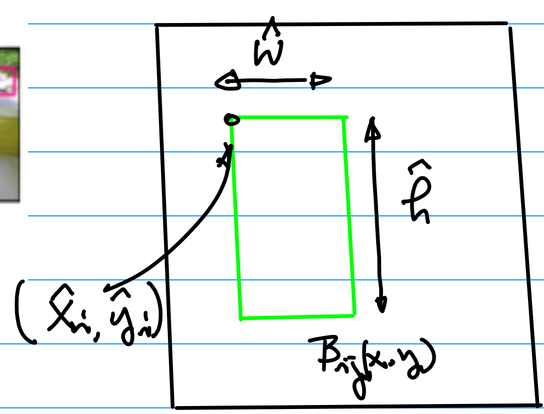


Fig1.a



$G_{ij}(x, y)$

Fig1b

CmpE258

Oct. 25, 22

Note: Midterm Exam Scheduled on the 3rd, Nov.

$$+\sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i\right)^2$$

$$+\lambda_{\text{noobj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i\right)^2$$

Loss function Based on the Confidence of Bounding Box

Loss function of the class probability.

Ground Truth

Estimated Probability

$$+\sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c\in\text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Prob(c)

Class, example, Traffic Signs, Pedestrian, Vehicles etc.

Note: To be able to Interprete & Apply this Technique to Yolo and other Related Design Need.

Example: Ref:

2022S-114c-Kmean-handCalculat…

2022S-114c-KmeanCluster-v3-20…

Background: A Tool to Create Probability Distribution Map. (In Fig 1a, pp 42)

2022S-114c-KmeanCluster-v3-2022-4-19.pdf

$$\underset{\mathbf{S}}{\arg\min}\sum_{i=1}^{k}\sum_{\mathbf{x}\in S_i}\left\|\mathbf{x}-\boldsymbol{\mu}_i\right\|^2 = \underset{\mathbf{S}}{\arg\min}\sum_{i=1}^{k}|S_i| \operatorname{Var} S_i \quad \ldots(1)$$

Feature vector $\vec{X} = (x_1, x_2, \cdots x_N)$ with Dimension N. $\rightarrow X$

for Example N=2.

a set of observations $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$,

Experiment 1

$X_1 = (x_{11}, x_{12})$

1st Sub: For the Experiment 1.

So, for K-th Experiment, we have $X_k = (x_{k1}, x_{k2})$, etc.

$\mu_i = (\mu_{i1}, \mu_{i2}, \cdots, \mu_{iN})$, mean of the given Class i

A Vector, Same dimension as the feature Vectors

for N=2, $\mu_i = (\mu_{i1}, \mu_{i2})$

i index for Class i.

CmpE258

Oct.25,22

44

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x}\in S_i} \underbrace{\|\mathbf{x}-\boldsymbol{\mu}_i\|^2} = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i \quad \cdots (1)$$

then,

$\sum_{x \in S_i}$ Take all feature Vectors X, as Long as they are from the Class S_i, Class i

Comparison, for the purpose of Classifying / Grouping a feature Vector to Class i.

Mean (AVG)

μ_i : feature Vector Distribution

Example: $X_j = (x_{j1}, x_{j2})$

$\mu_i = (\mu_{i1}, \mu_{i2})$

$X_i = (x_{i1}, x_{i2})$

x_2

x_1

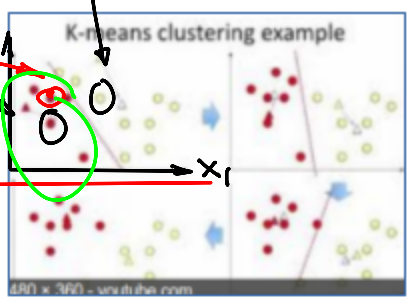


Fig.2

Cluster Seeking

Graphically μ_i a Clusster

features: Edge Comp. from Edge Detection; Color Distribution; Contours

Then, To make sure the minimization to Cover all classes ($\arg\min_S$):

$\sum_{i=1}^{K}$ Total K Classes.

$X - \mu_i$

↓

Minimize the Classification Error (Distance) By Grouping Feature Vector to its Right (Optimal) Cluster (Class)

"argmin" Or "Min" Minimization

↓

$\arg\min_S$ ← for All the possible Classes

Oct.27.

Midterm Exam is scheduled on Next Thursday (Nov.3rd). Brief Review Session will be conducted on next lecture.

Example: K-mean Calculation

2022S-114c-KmeanCluster-v3-2022-4-19.pdf

CmpE258

Oct. 27, 22

45

b. t Iteration

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \le \|x_p - m_j^{(t)}\|^2 \ \forall j, 1 \le j \le k\} \quad \ldots (2)$$

a. $\{S_i \mid i = 1, 2, \cdots, K\}$

Classes / Group.

if $\|\vec{X}_p - \vec{m}_i^{\,t}\|^2 < \|\vec{X}_p - \vec{m}_j^{\,t}\|^2$

$\vec{X}_p$ is better fitted into the Group i, O/W, $\vec{X}_p$ is better fitted to Group j

c. $\vec{X}$ feature vector

$\vec{m}_i^{\,t}$ ← Iteration, Updated @ step t

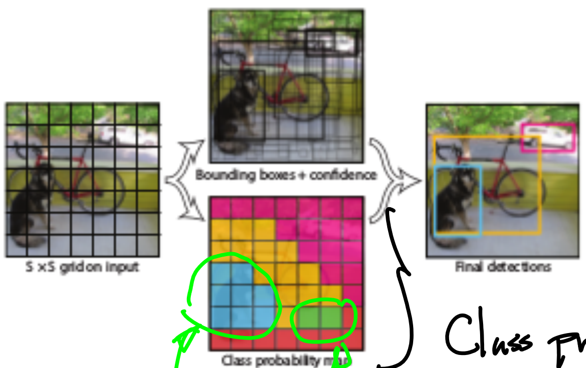
mean from Class i

↓

Condition(s) in Eqn (2).

$\forall j$, such that $1 \le j \le K$

For Any j j is in this Range, And it Covers all the groups. ($\because \sum_{i=1}^{K}$ from Eqn (1)).



Class probability map.

Probability of Class i

Probability of Class j

$$\vec{m}_i^{\,t} = \begin{pmatrix} m_{i1}^t \\ m_{i2}^t \end{pmatrix} \quad \ldots (3)$$

Take Eqn (2) to Actual Computation.

d. $\|\cdot\|^2$ Euclidean Distance

$$\|\vec{X}_p - \vec{m}_i^{\,t}\|^2$$

$$= \left\| \begin{pmatrix} x_{p1} \\ x_{p2} \end{pmatrix} - \begin{pmatrix} m_{i1}^t \\ m_{i2}^t \end{pmatrix} \right\|^2 = \left\| \begin{matrix} x_{p1} - m_{i1}^t \\ x_{p2} - m_{i2}^t \end{matrix} \right\|^2$$

$$= \sqrt{(x_{p1} - m_{i1}^t)^2 + (x_{p2} - m_{i2}^t)^2}$$

Update step: Calculate the new means to be the centroids

Algor
1).

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j \quad \ldots (3)$$

Vector for Group i, mean, e.g. Average;

"t+1" indicates it is updated from the previous Stage / step "t"

$$\frac{\vec{X}_1 + \vec{X}_2 + \ldots + \vec{X}_Q}{Q}$$

Q: Total Number of the feature Vectors from ith Group

$|S_i^t|$ Total Number

Compare 2 Groupings, e.g. 2 Ways of Classify feature vector $\vec{X}_p$ to Group i and Group j

CmpE258

Oct. 27, 22

46

Example: Hand Calculation

2022S-114c-Kmean-handCalculation1-converted.pdf.pdf

$x_1=\begin{bmatrix}0\\0\end{bmatrix}$ $x_2=\begin{bmatrix}1\\0\end{bmatrix}$ $x_3=\begin{bmatrix}0\\1\end{bmatrix}$ $x_4=\begin{bmatrix}1\\1\end{bmatrix}$

$x_5=\begin{bmatrix}2\\1\end{bmatrix}$ $x_6=\begin{bmatrix}1\\2\end{bmatrix}$ $x_7=\begin{bmatrix}2\\2\end{bmatrix}$ $x_8=\begin{bmatrix}3\\2\end{bmatrix}$

$x_9=\begin{bmatrix}6\\6\end{bmatrix}$ $x_{10}=\begin{bmatrix}7\\6\end{bmatrix}$ $x_{11}=\begin{bmatrix}8\\6\end{bmatrix}$ $x_{12}=\begin{bmatrix}6\\7\end{bmatrix}$

$x_{13}=\begin{bmatrix}7\\7\end{bmatrix}$ $x_{14}=\begin{bmatrix}8\\7\end{bmatrix}$ $x_{15}=\begin{bmatrix}9\\7\end{bmatrix}$ $x_{16}=\begin{bmatrix}7\\8\end{bmatrix}$

$x_{17}=\begin{bmatrix}8\\8\end{bmatrix}$ $x_{18}=\begin{bmatrix}9\\8\end{bmatrix}$ $x_{19}=\begin{bmatrix}8\\9\end{bmatrix}$ $x_{20}=\begin{bmatrix}9\\9\end{bmatrix}$

Nov. 1st (Tue).

Note: 1° Midterm Exam is scheduled on Nov. 3rd, Thursday. 1 hour, + 15 min for Submission.

1) 4:30pm ~ 5:30pm. plus 15 minutes Submission. 3 Questions

2) One page Formula Sheet is allowed, But has to be submitted once the exam is Concluded;

3) Complete the work independently. No Text messagings, No Discussion;

4) Question(s) on the Basic Theory Scope is defined By the Lecture Notes, White papers, Research Paper

(1) on Yolo, ppt;

Step 1. Basic Building Block of NN.

2022S-103a-notation-neuro-loss-function-2022-2-8.pdf

Feature Vector X, A Single Neuron, Weights $W=(w_1, w_2, \cdots, w_N)$, bias /a

Math Formulation of A Transfer function

$$h(X, W, b) = \sum_{i=1}^{N} w_i x_i + b$$

Activation Function

$$f(X, W, b) = f(h(X, W, b))$$

Formula

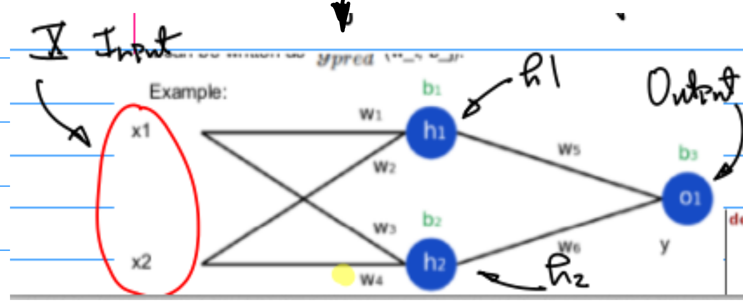
Block Diagram

physical meaning.

Step 2.

Feed Forward NN

- Math Formulation: PPT.
- Neural Network Architecture



Step 3. Move to Deep Convolutional NN. Pre-request is F.F.N.N.

a. Python Code Example

b. Concept of Loss Function

j-th Experiment

i-th Output Neuron

$y_i^j - \hat{y}_i^j$

CmpE258

Nov. 1st, 22

C. Learning/Training By Steepest Gradient Descent.

2022S-105c-#20-2021S-4gradient-descent-v2-final-2021-2-8.pdf

$$(x_1^{k+1}, x_2^{k+1}) = (x_1^k, x_2^k) + [-\eta(\nabla f)^t] \quad (5)$$

$$(x_1^{k+1}, x_2^{k+1}, \cdots, x_p^{k+1}) =$$

$$(x_1^k, x_2^k, \cdots, x_p^k) + (-\eta(\nabla f)^t)$$

Step size

Note: for ∇f Calculation;

justification → Derivate if Needed

$$f(x_1, x_2) - f(a, b) = -(f_{x_1}^2 + f_{x_1}^2) < 0 \quad (13)$$

Convolution: Hand Calculation 2D Conv.
K-by-K Kernel

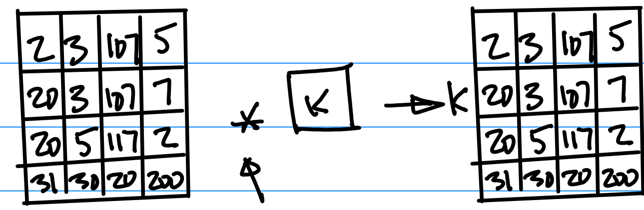
K odd Number "Un-Biased"

$$\frac{K-1}{2} + \frac{K-1}{2} = K-1$$ for Both Rows & Columns

Strides = 2
→ Resolution Reduction

Example: When K=1 for 1×1×C Convolution
Channel

Sol
Suppose C=1
1×1×1 kernel for 2D Convolution

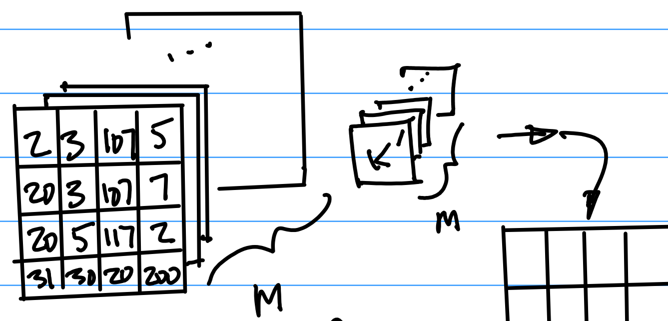


Convolution with 1×1×1 Kernel

Fig.1.

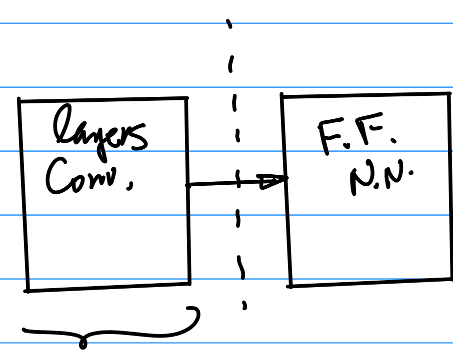
Suppose we have ~~feature~~ M feature planes

Fig.2



Channels of the feature Layers

Channel=1



Feature Extraction.

Step 4. MNIST & YoLo

Architecture, Activation Function
RELU, Softmax, Sigmoid.

CmpE258

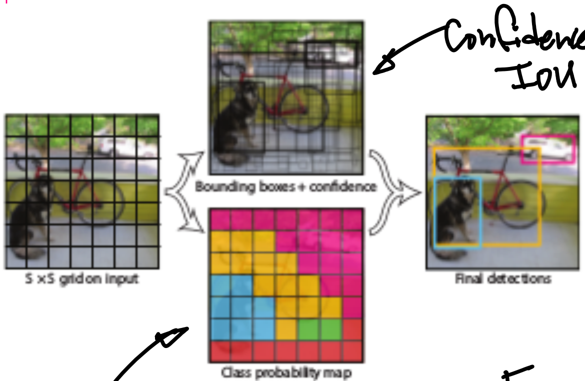
Nov. 1, 22. 48

T.F. Coding for MNIST

Theoretical Analysis for YoLo.

Eqn (1).

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}^{\text{truth}}_{\text{pred}} = \Pr(\text{Class}_i) * \text{IOU}^{\text{truth}}_{\text{pred}} \quad (1)$$



Confidence. IOU

Bayesian Theorem.

Fig. 1.

Projects & Homework Assignment.

Note: Preprocessing. (C.V.).

Binary Image Analysis &

Contours (Hand Calculation ~~& Coding)~~

Nov. 8 (Tue)

Continuation on K-mean Technique.

Example: Math. Formulation

2022S-114c-KmeanCluster-v3-2022-4-19.pdf

Hand Calculation

/ 2022S-114c-Kmean-handCalculation1-converted.pdf.pdf

The Approach:

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \le \|x_p - m_j^{(t)}\|^2 \ \forall j, 1 \le j \le k\}$$

Pre-requisit: Extract Feature Vectors.

$X_1=\begin{bmatrix}0\\0\end{bmatrix}$ $X_2=\begin{bmatrix}1\\0\end{bmatrix}$ $X_3=\begin{bmatrix}0\\1\end{bmatrix}$ $X_4=\begin{bmatrix}1\\1\end{bmatrix}$

$X_5=\begin{bmatrix}2\\1\end{bmatrix}$ $X_6=\begin{bmatrix}1\\2\end{bmatrix}$ $X_7=\begin{bmatrix}2\\2\end{bmatrix}$ $X_8=\begin{bmatrix}3\\2\end{bmatrix}$

$X_9=\begin{bmatrix}6\\6\end{bmatrix}$ $X_{10}=\begin{bmatrix}7\\6\end{bmatrix}$ $X_{11}=\begin{bmatrix}8\\6\end{bmatrix}$ $X_{12}=\begin{bmatrix}6\\7\end{bmatrix}$

$X_{13}=\begin{bmatrix}7\\7\end{bmatrix}$ $X_{14}=\begin{bmatrix}8\\7\end{bmatrix}$ $X_{15}=\begin{bmatrix}9\\7\end{bmatrix}$ $X_{16}=\begin{bmatrix}7\\8\end{bmatrix}$

$X_{17}=\begin{bmatrix}8\\8\end{bmatrix}$ $X_{18}=\begin{bmatrix}9\\8\end{bmatrix}$ $X_{19}=\begin{bmatrix}8\\9\end{bmatrix}$ $X_{20}=\begin{bmatrix}9\\9\end{bmatrix}$

Note: Human Expertise Knowledge, Heuristics, Can be Applied to Perform inspection of Feature Vectors. which leads to the following Assumption: K=2

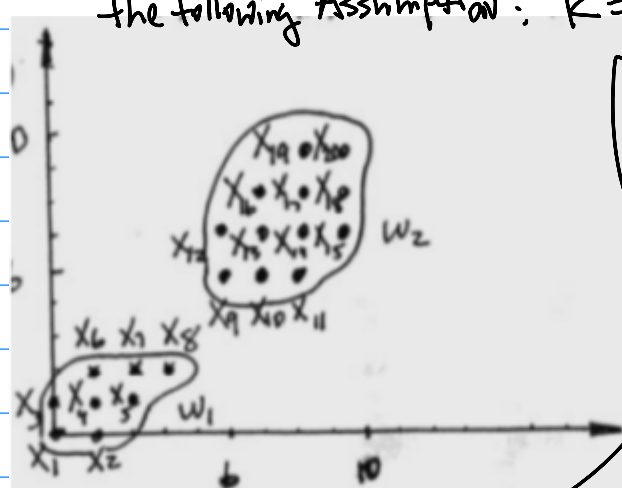


Fig. 2

The Class Number = 2.

Step1. Define Number of Cluster $K=2$ Based on Heuristics (the plot of the feature vectors).

Let $K=2$

And make initialization by arbitrarily Select 2 points $\vec{x}_1$, $\vec{x}_2$ as the cluster Center

Note: Assume $K=2$ reflects Human Expert Knowledge.

Initialize $\vec{m}_i$

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Hence, we have

Let cluster

$$\vec{m}_1^0 = \vec{x}_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \cdots (1)$$

$$\vec{m}_2^0 = \vec{x}_2 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \cdots (2)$$

We Can Randomly / Arbitrarily Assign Each feature Vector $\underline{X}$ to each Class,

And Let Class 1 be

$$C_1 : (\vec{x}_1, \vec{x}_3) \quad \cdots (3)$$

class 2 be

$$C_2 : (x_2, x_4, x_5, \cdots, x_{20}) \quad \cdots (4)$$

Step2, Use Equation

$$\|\vec{x}_p - \vec{m}_i\| \leq \|\vec{x}_p - \vec{m}_j\| \quad \cdots (5)$$

for $\vec{x}_p$ from Class i to regroup feature Vectors in Class i to j

Note: Compare the distance "$\|.\|$" Resulted from A Feature Vector $\underline{X}_p$ belongs to class i to Any other Classes j.

$j = 1, 2, \cdots, K$ And $j \neq i$

From Class C_1, Leftside of Eqn.(5)

$$\|\vec{X}_1 - \vec{m}_1^0\| = \left\| \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} - \begin{pmatrix} m_{11} \\ m_{12} \end{pmatrix} \right\| = \qquad m_{11} = x_{11}, \ m_{12} = x_{12}$$

$$= \sqrt{(x_{11} - x_{11})^2 + (x_{12} - x_{12})^2} = 0$$

the Right Hand Side of Eqn(5)

And
$$\|\vec{x}_1 - \vec{m}_2\| = \left\| \begin{pmatrix} x_{11} \\ x_{12} \end{pmatrix} - \begin{pmatrix} m_{21} \\ m_{22} \end{pmatrix} \right\| = \qquad m_{21} = x_{21}, \ m_{22} = x_{22}$$

$$= \sqrt{(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2}$$

$$= \sqrt{1^2 + 0^2} = 1$$

Therefore, Eqn(5) Holds good. Hence, $\underline{X}_1$ belongs to C_1.

Continue the process, with the next feature Vector.

Take feature Vector $\vec{X}_3$.

$\| \vec{X}_3 - \vec{m}_1^0 \| = \sqrt{(X_{31}-X_{11})^2 + (X_{32}-X_{12})^2}$

$= \sqrt{0^2 + 1^2} = 1$, Left Hand Side of Eqn(5)

$\| \vec{X}_3 - \vec{m}_2^0 \| = \sqrt{(X_{31}-X_{21})^2 + (X_{32}-X_{22})^2}$

$= \sqrt{1^2 + 1^2} = \sqrt{2}$, Right Hand side of Eqn(5)

Hence,

$\| \vec{X}_3 - \vec{m}_1 \| \le \| \vec{X}_3 - \vec{m}_2 \|$

so $\vec{X}_3$ stays in C_1.

Continue the process,

Therefore $\{\vec{X}_1, \vec{X}_3\} \in C_1$, and $\{\vec{X}_2, \vec{X}_4, \vec{X}_5, \cdots, \vec{X}_{20}\} \in C_2$

Step 3. Update Clusters

$\vec{m}_1^1 = \frac{1}{N}\sum_{i=1}^{N} \vec{X}_{1i} = \frac{1}{2}(\vec{X}_1 + \vec{X}_3)$

$= \begin{pmatrix} 0 \\ 0.5 \end{pmatrix}$

$\vec{m}_2^1 = \frac{1}{18}(\vec{X}_2 + \vec{X}_4 + \vec{X}_5 + \cdots + \vec{X}_{20})$

$= \begin{pmatrix} 5.67 \\ 5.33 \end{pmatrix}$

Now, Continue the process till the Convergence Reached, e.g., there is No New Classification, (No New Grouping). Also, this means the Clusters $\vec{m}_i$ $(i = 1, 2, \cdots, k)$ Stay the Same.

Step 4. Check the clusters for Convergence verification.

$\vec{m}_1^0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$, $\vec{m}_1^1 = \begin{pmatrix} 0 \\ 0.5 \end{pmatrix}$,

Superscript: Indicates the Step of the iterations.

Since $\vec{m}_1^0$ and $\vec{m}_1^1$ are different, so the process continue

Step 5, with New updated Cluster, Compute

$\| \vec{X}_\ell - \vec{m}_1^1 \| \le \| \vec{X}_\ell - \vec{m}_2^1 \|$

for $\ell = 1, 2, \cdots 8$; (For Simplicity, the Steps of these Computation were not listed here, they are Similar to what we did), and

$\| \vec{X}_\ell - \vec{m}_2^1 \| \le \| \vec{X}_\ell - \vec{m}_1^1 \|$

for $\ell = 9, 10, \cdots, 20$;

$\Downarrow$

Regrouping Feature Vectors, Leads to

$C_1 = \{X_1, \cdots, X_8\}$, $C_2 = \{X_9, X_{10}, \cdots, X_{20}\}$

Now, Update means (clusters)

$\vec{M}_1^2$, $\vec{M}_2^2$

$$\vec{m}_1^2 = \frac{1}{8}(\vec{X}_1 + \vec{X}_2 + \cdots + \vec{X}_8) = \begin{pmatrix} 1.25 \\ 1.13 \end{pmatrix}$$

$$\vec{m}_2^2 = \frac{1}{12}(\vec{X}_9 + \vec{X}_{10} + \cdots + \vec{X}_{20}) = \begin{pmatrix} 7.67 \\ 7.33 \end{pmatrix}$$

Check for the convergence,

$\because \vec{m}_1^2 \neq \vec{m}_1^1$, $\vec{m}_2^2 \neq \vec{m}_2^1$

Not Converged, Continue

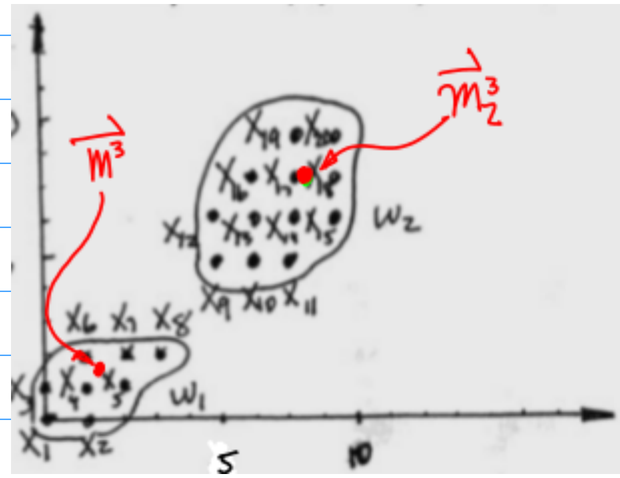
Therefore, we will Continue the Computation. we go through the Same process, which Leads $\{\vec{X}_1, \vec{X}_2, \cdots, \vec{X}_8\}$ Same as the previous Class, And $\{\vec{X}_9, \vec{X}_{10}, \cdots, \vec{X}_{20}\}$ is also the Same as the previous Class. which gives

$\underline{\vec{m}_1^3 = \vec{m}_1^2}$ and

$\underline{\vec{m}_2^3 = \vec{m}_2^2}$

Now, the clusters Converged Found the Cluster Center:

$$\vec{m}_1^3 = \begin{pmatrix} 1.25 \\ 1.13 \end{pmatrix}, \quad \vec{m}_2^3 = \begin{pmatrix} 7.67 \\ 7.33 \end{pmatrix}.$$



Note: K-mean in OpenCV → Color Segmentation

Probability Distribution Map.

K-mean Can also be Applied for other feature Vectors Rather than just image Color. → Sometimes Need Dimension Reduction

↓

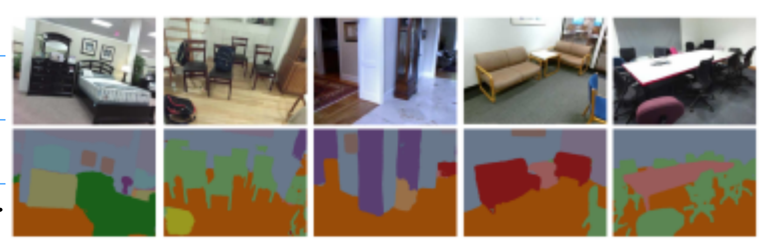
Principle Componment Analysis.

Now, Discussion On Semantic Segmentation. //

Bounding Box Based Object Recognition (YOLO) → Pixel Based Object Recognition

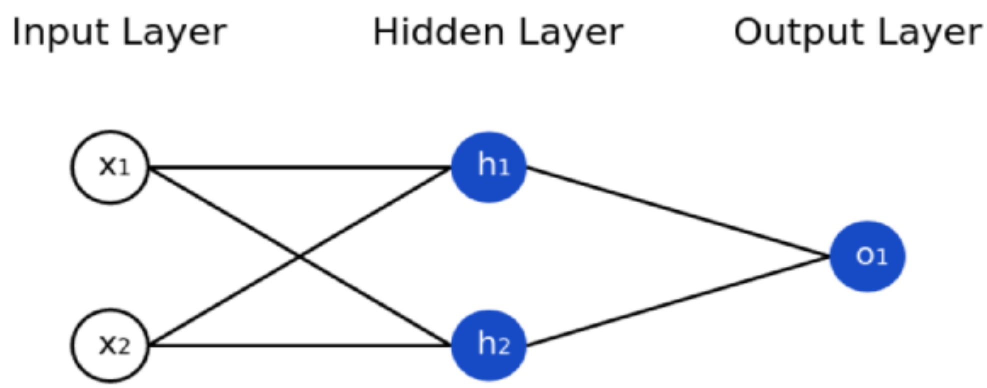
Example:

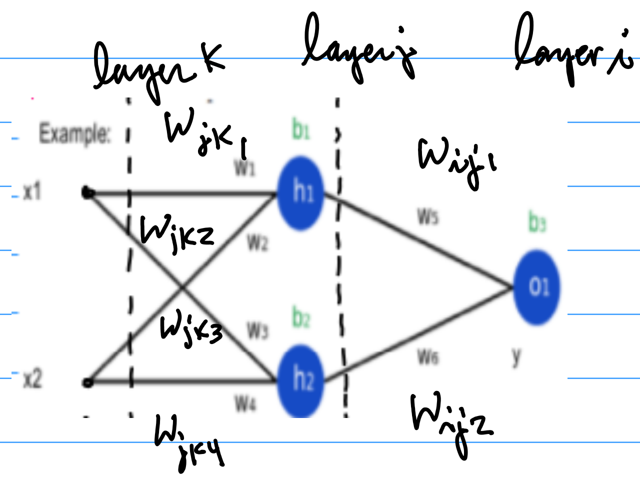
Before v.s. After.



Feed Forward NN

https://victorzhou.com/blog/intro-to-neural-networks/





$$(w^{k+1}_{ij1}, w^{k+1}_{ij2}) = (w^{k}_{ij1}, w^{k}_{ij2}) + (-\eta(\nabla f)^{t})$$

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial w_{ij1}} \\ \frac{\partial f}{\partial w_{ij2}} \end{pmatrix}$$

Where $f = \frac{1}{2}\sum_{j=0}^{M-1}\sum_{i=0}^{N-1}\left(y_i^j - \hat{y}_i^j\right)^2$

loss function

And y_i^j is the Output

$$y = f\left(\sum_{i=1}^{N} w_i x_i + b\right) = f(h(w_i x_i; b))$$

y_i^j ith output @ the last layer.
@ Experiment j

Drop j. Rewrite Subscript.

Range them from Output Layer to the input layer.

$W_{ij} \to W_{jk}$, Scalable

$$(x_1^{k+1}, x_2^{k+1}) = (x_1^k, x_2^k) + [-\eta(\nabla f)^t]$$

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \dots \\ \frac{\partial f}{\partial x_i} \\ \dots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

$$y_{ij} = f\left(\sum_k w_{ijk} x_{ijk} + b_{ij}\right)$$

$$= f\left(h\left(w_{ijk} x_{ijk}, b_{ij}\right)\right)$$

$$\frac{\partial}{\partial w_{ijk}} L = \frac{\partial}{\partial w_{ijk}} \left(\frac{1}{2} \sum_{\mu} \sum_{k} \left(y^{\mu}_{ijk} - \hat{y}^{\mu}_{ijk} \right)^2 \right)$$

$$\rightarrow \underbrace{2 \cdot \frac{1}{2} \cdot \left(\sum_{\mu} \sum_{k} \left(y^{\mu}_{ijk} - \hat{y}^{\mu}_{ijk} \right) \right)}_{\delta} \frac{\partial}{\partial w_{ijk}} \left(- y^{\mu}_{ijk} \right) = -\delta \frac{\partial}{\partial w_{ijk}} f\left(\sum_k w_{ijk} x_{ijk} + b_{ij} \right)$$

$$= -\delta f'(\cdot) \frac{\partial}{\partial w_{ijk}} \left(\sum_k w_{ijk} x_{ijk} + b_{ij} \right)$$

$$= -\delta f'(\cdot) \frac{\partial}{\partial w_{ijk}} \sum_k w_{ijk} \quad = -\delta f'.$$