20170243 심재원

$P(A|B) = \frac{P(A,B)}{P(B)}$ $P(B|A) = \frac{P(A,B)}{P(A)}$

1. Bayes Theorem

$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$

disease infects only one in a hundred people → $\frac{1}{100}$

You tested this disease and test was positive.

99.8% accurate → $\frac{998}{1000}$

1-a. positive result (given) you have disease

$\frac{1}{100}$

1-b. positive result (given) you don't have disease.

$\frac{99}{100}$

1-c. positive result.

$\frac{1}{100} \cdot \frac{1}{2} = \frac{1}{2}$

1-d. disease (given) the test was positive.

$\frac{1}{2} \cdot \frac{998}{1000} = \frac{499}{1000}$.

2017024?3 심재윤

#2. Corner detection.

$$E(u,v) \approx \begin{bmatrix} u \\ v \end{bmatrix}^T M \begin{bmatrix} u \\ v \end{bmatrix}$$

#2-a.

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2(x,y) & I_x(x,y) I_y(x,y) \\ I_x(x,y) I_y(x,y) & I_y^2(x,y) \end{bmatrix}$$

#2-b.

- A patch contains an edge if ①
- A patch contains a corner if ③
- A patch is flat if ④

#2-c.

Corner response score : $R = \det(M) - k \cdot tr(M) = \lambda_1 \lambda_2 - k \cdot (\lambda_1 + \lambda_2)^2$

(k is constant 0.04~0.06)

20170243 심재윤

#3.

#3-a. How interest points are detected in SIFT using Gaussian kernel.

→ Gaussian filtered image를 input 이미지 들로부터 구하고 인접한 Gaussian filtered image들을 빼면 DoG filtered image를 얻게 된다. 이과정에서 blob score를 계산하고 interest point인 blob을 locally maximum score 를 가지게 Non maximum suppression을 통해 찾을 수 있다.

#3-b. why SIFT is invariant to scale and rotation.
scale-space search 중에

→ SIFT는 DoG filter를 사용하게 되는데 이는 LoG filter와 (k-1)배 비례 관계에 있으며, LoG filter는 σ^2를 곱해줌으로서 Scale에 invariant 해졌다. 따라서 DoG도 Scale에 자연스럽게 invariant 해지고, rotation 같은 경우는 dominant orientation을 estimate 하기에 invariant 하다.

#3-c.

- False.

- True.

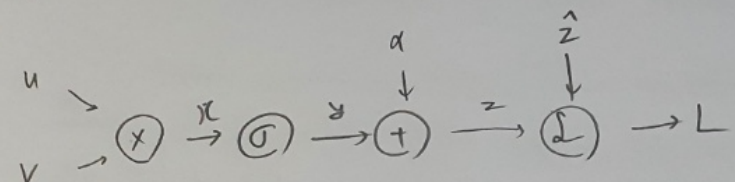
- True.

- Bag of visual words histograms captures spatial configuration of image.

→ False.

20170243 정재윤

#4.



$L = \mathcal{L}(\sigma(uv)+\alpha, \hat{z})$, where $\mathcal{L}(x,y) = \frac{1}{2}(x-y)^2$, $\sigma(x) = (1+e^{-x})^{-1}$, $\frac{d\sigma}{dx} = \sigma(x)(1-\sigma(x))$

$$\begin{cases} uv = z \\ y = \sigma(x) = (1+e^{-x})^{-1} \\ y+\alpha = z \\ \mathcal{L}(z,\hat{z}) = L = \frac{1}{2}(z-\hat{z})^2 \end{cases}$$

$L = \mathcal{L}(\sigma(x)+\alpha, \hat{z})$

$= \mathcal{L}(y+\alpha, \hat{z})$

#4-a $\frac{\partial L}{\partial z} = \frac{1}{2} \cdot 2 \cdot 1 = 1$

#4-b $\frac{\partial L}{\partial \alpha} = \frac{\partial L}{\partial z} \cdot \frac{\partial z}{\partial \alpha} = 1 \cdot 1 = 1$

#4-c $\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z} \cdot \frac{\partial z}{\partial y} = 1 \cdot 1 = 1$

#4-d $\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z} \cdot \frac{\partial z}{\partial y} \cdot \frac{\partial y}{\partial x} = 1 \cdot \left(-(1+e^{-x})^{-2} \cdot (-e^{-x})\right) = e^{-x}(1+e^{-x})^{-2}$

#4-e $\frac{\partial L}{\partial u} = e^{-x}(1+e^{-x})^{-2} \cdot \frac{\partial x}{\partial u} = e^{-x}(1+e^{-x}) \cdot v$

#4-f $\frac{\partial L}{\partial v} = e^{-x}(1+e^{-x})^{-2} \cdot \frac{\partial x}{\partial v} = e^{-x}(1+e^{-x})^{-2} \cdot u$

2017.02.13 성재원

5.

#5-a which operations in neural networks are accelerated by GPU?

대부분 deep learning 다 병렬 연산을 한다.

특히 대량의 convolution back propagation 등이 GPU를 필요로한다.

#5-b. size of dataset very small & network over fitted. what can you do to allieviate the issue?

① Drop out - 처음 두 FC layer 전체에 0.5의 확률로 뉴런을 0으로 처리할 것이다.

② weight decay - 이러면 네트워크 parameter 들이 가질 수 있는 값의 범위가 제한됨.

③ Early stopping - 중간 성능에서 일찍 training을 멈춘다.

#5-C. How gradient can be safely propagated from higer to lower layers?

gradient vanishing 문제를 해결하면 network를 깊게 진행하여 기울기 소실이 해결될 것이다. 가장 먼저 할건 weight initialization을 진행할 것 같다.

ReLU도 사용하고 Batch normalization도 사용할 것이다.

2018.0243 심재윤

#6.-a.

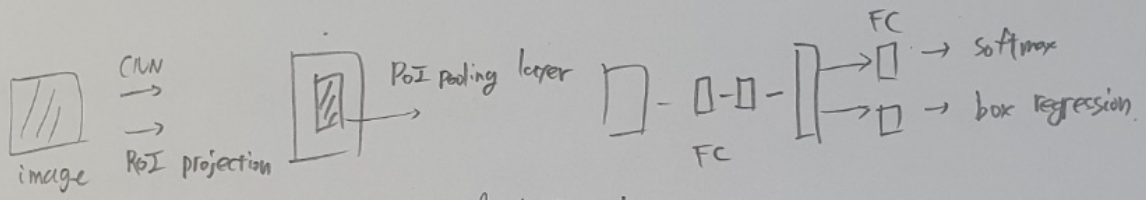
① wrong (no selective search in Faster R-CNN)

② wrong (the margin will be largest, that's the goal)

③ okay.

④ wrong (we need more negative samples)

#6-b.



single feature computation & RoI pooling.

objectives : softmax, box regressor

$L = -\log\left(\frac{\exp(y_i)}{\sum \exp(y_j)}\right)$ ← (classification loss, regression loss)
분류의 목적 BB의 좌표 예측의 목적.

#6-c. why the transformation parameters predicted for BB refinement has to be invariant to the scale of box proposal. and how achieve it?

→ $d(P') = (dx, dy, dw, dh)$

training에서 가까운 ground truth Box를 매치시킬 것이다.
그럴려면 transformation parameter를 예측하는데 이러한 proposal이 각 ground truth에 얼마큼 근사할까.

2017024-3 심재윤

#7.

#7-a. limit of FCN and how DeconvNet solve

receptive field가 고정된 사이즈이다. 그리고 upsampling 과정이 단순하다. 이러한 한계들은 Convolutional encoder-decoder 인 DeconvNet이 해결했다. 이 구조는 원래 이미지 해상도에서 거의 정확한 모양을 되찾을 수 있으며 end-to-end 방식으로 더 나은 결과를 낼 수 있다.

#7-b. main challenge in end-to-end receptive field size ⇑

① network를 깊게 쌓는다, pooling layer를 많이 쌓게 한다.
② unpooling & deconvolution
③ Batch normalization
④ skip connection

#7-c. difference region pooling layers Fast R-CNN & Mask R-CNN.

Mask R-CNN에서는 RoI Align을 적용했다. 그리고 Faster R-CNN에서 RoI에 mask를 적용해 각 픽셀이 객체인지 아닌지 결정했다.

2017043 심재원

#8-a. Sol: 이 데이터 들을 kernel method를 사용해 더 고차원의 공간에 mapping해준다.
Gaussian kernel ($\exp\left(\frac{-|x_i - x_j|^2}{\sigma^2}\right)$) 을 사용하면 좋을것 같다.

#8-b. 앞에서 언급한 linear kernel ($x_i^T x_j$) 나 polynomial kernel ($x_i^T x_j)^d$ 보다 더 효과적으로 나눌수 있을것 같다.

#8-c. hidden layer는 데이터를 더 정교하게 나뉘주는데 도움이 된다.