lec 11 Image classification / /

Image classification challenges
① viewPoint Variation ② Interclass ③ fine grained Categories
④ Background clutter ⑤ illumination ⑥ Deformation ⑦ occlusion

Applications
① object detection ② disease diagnosis ③ Image Captioning
④ playing Games (ex: Go) ⑤ Activity analysis

Image classification Based on Data driven approach
① Collect dataset of imgs, labels
② use Machine learning to train classifier
③ Evaluate Classifier to new images

Dataset
① MNIST → handwritten numbers → 10 classes → 28X28
② CIFAR → 100class → 32X32 → 20 Superclass
③ ImageNet → 1000 class top5 accuracy algo
④ MIT places → 365 Classes → 256 X 256

Conventional ML Classifier (Classification over level of features)
① K-nearest neighbour (kNN) ⇒ Not pixel level
↓
train: Memorize all data labels
test: predict label of most similar training image
uses
distance metric to Compare imgs
ex L_1 distance ⇒ $\sum_p |I_1 - I_2|$

Decision boundry → boundry between 2 classification regions

Can be noisy, affected by outliers

(kNN)

↳ instead of Copying label from nearest neighbour
↳ take Majority vote from K closest points

use more neighbours → smooth out region decision boundries, help to reduce effect of outliers

when $k > 1$ ⇒ ties between classes

⇒ with right Choice of distance metric, we Can apply knn to any type of data

(L1) → manhattan distance $|I_1^p - I_2^p|$ (Modulus)

(L2) → Eclidean distance (root) $\sqrt{\sum_p (I_1^p - I_2^p)^2}$

knn - hyper parameters → (k)
→ distance metric
} Very Probla Dependant

knn → (X) pixels but with ConVNet Features works well

Before → linear score function → $F = wx$

Now → non linear (2-layer Neural net) → $F = w_2 \max(0, w_1 x)$

/ /

Fully Conected Neural Network (او) Multi layer perceptron
↳ عشان كل نيورون موجود في layer متصل بنيورون في الـ layer اللي بعدها

↓

ex Back Propagation model (Forward model)
activation function ⇒ Sigmoid

(1) Forward Compute output
Backward Compute gradients

Deep Neural Network (generalized form from MNN)

(2) relu activation functions
hirerachial learning algo with many hidden layers
↳ اختار rectified linear unit

⇒ if we build neural network without activation Function
فيبقى ⇒ linear classifier

Activation Function examples
(1) Sigmoid (2) tanh (3) Relu (4) leaky Relu (5) MaXout (6) elu
احسنهم

Convention Neural network (CNN)

1. Fully Connected layers → output يمثل
2. Activation function → relu (max(0, wx))
3. Convolution layers
4. pooling layers → redule dimensions
5. Normalization

1. Fully Connected layer

input array → stretch to vecter → X weight → dot product between row of (w), input

2. Convolutional layer (learn feature) stack 2 Convolution → get another Convolution!

3X32X32 Image — Filter extend full depth ch input volume

3X5X5 Filter ⇒ Convolve filter with img ⇒ slide over img spatially to Compute dot product

in first Convolutional layers → local image templates (simplest features)

Receptive Fields

Problem: large image need many layers
Solution: Downsample inside the network

3. Pooling layer ⇒ reduction of features

downsampling

maxPool → maximum

kernel size — stride — Pool function

Batch Normalize

Idea → Normalize outputs of layer so, they have Zero mean = 0, unit Variance = 1

↓

reduce intervel Covariate Shift

Batch Normalization
Fully Connected

X : N x D

Convolutional
Batch normalization

X : N x C x H x W

Batch Normalization
usually inseted after Fully Connected layer or Convolutiond layer

before non linear

✓
much easy to train
high learning rate
Fast Convegale
robust to intialization

✗
- Not well undestood
- Behave using training,