Date:

Convolution Neural Network

Sun Mon Tue Wed Thu Fri Sat

~~ANN vs CNN~~:

CNNs:

- Sparse connectivity
- Shared weights → reduced parameters.
- Captures spatial hierarchies
- Automatically learn hierarchial features from images
- Efficiently process larger images
- Superior performance on image related tasks

Consist of: Interconnected neurons, Convolutional layers, pooling layers

edges → combination of edges / shape → objects.

Building blocks:

- Convolutional Layers: Apply filters to extract features from input data
- Padding: Adjust spatial dimensions of the output volume to preserve resolution.
- Batch normalization: Normalize inputs of each layer to improve training speed & stability

Date:

Sun Mon Tue Wed Thu Fri Sat

- Stride: Determine the step size at which filters convolve.
- Pooling layer: Reduce spatial dimensions of the inputs while retaining information.
- Dropout layer: Randomly deactivates neurons during training to prevent overfitting.
- Fully connected layers: Connect every neuron in one layer to every neuron in next layer. for classification/regression
- Activation function: Introduce non-linearity to enable the network to learn complex features

Convolution:

- feature extraction
- Sliding filter → kernel.
- ~~filter~~ ~~replaced features~~
- Summition (filter value × pixel value) → one value of feature map.

- Different filter size = Different levels of abstractions.

- filters start with random values,
- Adjusted using backpropagation to reach optimal filters

for classification using deep Network:

- feature maps is flattened & passed to NN.

Date:

Sun Mon Tue Wed Thu Fri Sat

Spatial Exploitation based CNNs:

- Small filter → fine grained
- large sized filter → coarse grained.

LeNet: 1998:

- first CNN → traditional NN
- Considered each pixel as an input which increase the computing
- Neighboring pixels are correlated.
- LeNet used ~~this~~ to convolution filters
- filters shared the learning across the image.

- Didnt reduce the computations but learned features from raw images

- 5 layers 60k to 70k parameters 5x5 filters

Conv - Pool - conv - Pool - flatten - fc - fc

Date:

Sun Mon Tue Wed Thu Fri Sat

AlexNet (2012)

- first deep CNN
- was overfitting cuz of more layers so:
  1- Relu was used to improve convergence by removing vanishing gradient.
  2- some transformational units were skipped Randomly
  3- overlapping subsampling & local response normalization

5×5 filters 8 layers 60-70 million parameter.

Convo - max pool - Norm - Convo - max pool - Norm - Convo - Convo - Convo - Max pool - fc - fc - fc

Input = 227 × 227 × 3 $1 + \frac{\text{inputsize} - \text{kernel size} + 2(\text{padding})}{\text{Stride}}$

Convo L1: 96 units 11×11 filter

Strid = 4

$\frac{227 - 11}{4} + 1 = 55$ 55 × 55 × 96

Parametres: (11×11×3) × 96 = 35K

Pooling layer preserves feature map depth which means depth grows at every layer.

Date:

Sun Mon Tue Wed Thu Fri Sat

Zfnet: 2013:

- understand by represent the performance of CNN
- we start with output of a convo layer instead of an input image.
- ~~use applying pooling~~
- The reverse order of convo by pooling layer helps to convert an output to an image. This helps us understand which parts of input image are responsible for neuron activations.

- Experiments on AlexNet showed neurons in 1st by 2nd layers were ~~dead~~ dead
- They found distortions by errors in 2nd layer output.

- They minimized the learning o

6e7 - 70 million parameters

Reduces number channels in bottleneck layer

Date:

Sun Mon Tue Wed Thu Fri Sat

VGG: 2014:

- Showed the relation of depth with representation capacity of NN
- Used small filter size
- Small filter mean low computational complexity
- Regulates complexity by placing 1×1 conv filter between layers
- Max pooling by padding

138 million parameters – 16 layers – 3×3

143 million parameters – 19 layers – 3×3

GoogleNet 2014:

- AKA Inception V1
- Reduce computations by improve accuracy.
- Introduced inception block. Split, transform by merge.
- Different size filters. / Parallel convolution paths.
- Learns diverse types of variations present in same category of images.
- Regulates using 1×1 convolayer.
- Sparse connections.
- Omit feature maps that were irrelevant

Date:

Sun Mon Tue Wed Thu Fri Sat

- Global avg pooling in last layer
- Batch Normalization, RMSProp optimizer.
- No fc layers
- Introduced auxiliary learners.
- Needs to be customized from module to module.
- Due to bottleneck useful information is also lost.
-

5 million parameters.

Summary:

	Parameter	Layers	Filter	Stride
→ Lenet:	60K-70K	5(2c, 3fc)	5×5	1
→ AlexNet:	60M-70M	8(5c, 3fc)	1- 11×11 2- 5×5 3&5- 3×3	4 c1 1 c2,3,4,5
→ Zfnet:	60M-70M	8(5c, 3fc)	7×7 3×3	2 c1 1 c2,3,4,5
→ & VGG 16	138M	16(13c, 3fc)	3×3	1
→ VGG 19	145M	19(16c, 3fc)	3×3	1
→ GoogleNet	5M	22(20, 2FC)	1×1 3×3 5×5	1

Date:

Sun Mon Tue Wed Thu Fri Sat

Depth Based CNNs:

- Emphasize that increasing the depth of network will capture hierarchical representation of input data
- ~~Deep~~ Deeper network can learn more complex features
- -> Universal approximation theorem : States a single layer with infinite neurons can approximate any function.
- Deeper network can maintain expressive power at reduced cost
- Problems: 1- Slow training 2. convergence speed

Highway Networks: 2015

- gating mechanism to control ~~flow~~ information flow.
- Multipath CNN, cross layer connectivity
- 50 layers
- Normal CNN performance decrease even with 10 hidden layer
- Highway Network performed significantly better even with 900 layers.
- Gate control flow of information, when to pass info by when to carry.

Date:

Sun Mon Tue Wed Thu Fri Sat

forward propagation: gates determine how much information to flow through. gates are implemented as sigmoid functions. (output 0 & 1)

Backpropagation: gradients of loss of all parameters are calculated including gate weights.

- Network learns to adjust gate weights such that it effectively controls the flow of information.

→ Transform Gate: Decides how much of the input should be transformed. Allows a larger portion of input to pass when data contain complex features

→ Carry Gate: Determines how of input should be carried over to next layer without any transformation. when input features are already very informative. data contain important features that should be preserved

- Both gates are learned during training

Date:

Sun Mon Tue Wed Thu Fri Sat

ResNet: 2015

- Residual Learning
- 152 layers and still less computational complexity than any previous Network.

- 50/101/152 layer has less error than 34 layer plain Net.

Inception V3, V4 by Inception ResNet.

- Improved versions of inception V2, V2.
- V3 → Reduce computational cost
  - used 1x7 and 1x5 ~~filt~~ filter by 1x1 convolution for bottleneck.

- Combined Residual learning by inception blocks filter concatenation replaced with residual connection.

- Inception Resnet has more width by depth than normal Inception V4.

- Inception ResNet converges faster
- Residual connection accelerates training.

Auxilary classifier: Added classifier in the middle of network to improve performance

Stem: Initial operations

Date:

Sun Mon Tue Wed Thu Fri Sat

→ GoogleNet : 2014 22 layers 5 million parameters High comp

Inception v1: - Modules with multiple filter sizes.

→ Inception V2: 2015 42 layers 11 million parameters Mod-high comp

- Batch Normalization
- factorized 7x7 filters.

→ Inception V3: 2015 48 layers 23 million parameters Mod-High comp

- Auxilary classifiers
- Improved inception modules
- factorized 7x7 filters

→ Inception V4: 2016 ~~48~~ 76 layers ~~23~~ 42 million parameters High comp

- Stem with 3x3 convolutions.
- improved architecture

→ Inception ResNet V2: 2016 164 layers 55 million parameters High comp

- Utilizes residual connection
- improved architecture.

* Residual blocks: decrease spatial dimension of feature maps

- This helps to extract higher level abstract features
- condense feature maps

- Down sample to manage computation complexity.

- Pooling, 1x1 conv, big stride

Date:

Sun Mon Tue Wed Thu Fri Sat

Multipath Based CNN:

- Multipath or shortcut connections connects one layer to next another by skipping some intermediate layers to allow the specialized flow of information across layers

- Solves vanishing gradient problem

- Zero padded, projection based, dropout, skip connections 1×1 connections

- Multiple path with different receptive fields

Highway networks:

- gating mechanism
- allows training many layers (even 900layer)

when gate is open input is passed to next layers
when gate is closed then input is assigned one next layer (skipped layer)

ResNet:

- Residual connection.
- Adds input to output of layer

Depth problems: Time intensive, feature reuse, & vanishing / exploding gradient

fraction of improvement require many layers

Date:

Sun Mon Tue Wed Thu Fri Sat

DenseNet 2017:

- Densenet connectivity between layers
- Each layer connected to every other layer in feed forward
- Concatenates feature maps across

Widht based: Multic Connection CNN:

- Width is also essential parameter
- width defines the learning power of NN.
- Deep Networks some units weren't learning so they shifted ~~focus~~ to wide networks
- Increase number of channels of ~~in~~ a layer

Wide ResNet 2016:

- Some blocks in deep Networks were not contributing enough
- introduced additional factor k to increase widht
- Twice the number of parameters still than ResNet But can be better than deeper networks
- Dropout between layers rather than in Residual blocks

- 50 layer

Date:

Sun Mon Tue Wed Thu Fri Sat

PyramidalNet 2017:

- In deepl networks the depth of feature maps increase while the spatial dimensions decrease cuz of pooling layers. (down sampling)
- loss in spatial dimensions limits the learning ability
- In Pyramidal net width is increased gradually per ~~across~~ ~~existed~~ residual unit
- enables to cover all possible locations.
- uses 2 types of widening:
  1- additive: increases linearly
  2- multiplicative: increases geometrically.

- Pyramidal pooling layers: feature maps pooled at multiple scales to capture global & local information

- Multiscale feature fusion: fuse features from different scales using skip connections.

- Hierarchical Architecture: progressively learns features at different levels of abstractions
  low level → more abstract

- Shared parameters & parallel processing so more efficient

Date:

Sun Mon Tue Wed Thu Fri Sat

XceptionNet 2017:

- extreme inception architecture
- Modified orignal Inception block by making it wider
- replacing different scales by single 3x3 scale by 1x1 convo

- Replaces stndard convolution with depht wise convolution
  - depthwise convolution: operate on each input channel seprately applying single ~~feature~~ filter per channel.
  - point-wise: combines the outputs of depth wise allowing for cross channel interaction.

- More computationally efficient

- No intermidiate pooling layers, directly connects input to each convolution operation

ResNext: 2017: Aggeregate Residual Transformation Network

→ Number of paths.

- Cardinality: additional dimension, size of the set transformations
- Used :. deep homogenous topology of VGG
  - Simplified GoogleNet architecture (split, transform, merge)
  - Residual learning.

feature map wise statistic / fm motifs / fm Descriptor / channel wise descriptor
all same (output of Squeeze operation)

Date:

Sun Mon Tue Wed Thu Fri Sat

Feature Map Exploitation Based CNN:

- Some feature maps impart little-to no role in object discrimination.
- Some feature sets may create noise which leads to overfitting
- Selection of feature maps can help in improving generalization
- Compress or enhance feature maps.

Squeeze & exictation: 2018:

- Squeeze operation: reduce the spatial dimension of the input feature map to a single spatial point. (global-avg pooling)
  Spatial information → channel wise descriptor.
- Exictat operation: learns a set of weights that capture importance of each feature map. channel.
  - learn it using a small fully connected network
  - The weights are then applied to each feature map channel to modulate its importance

feature Re-calibrate: By performing squeeze & excite ops SENet recalibrate feature maps in a channel wise manner.

- This allows it to focus on informative features while suppressing less relevant ones

Date:

Sun Mon Tue Wed Thu Fri Sat

Integration with CNN: can be added as a ligut weight module after convolayers
- Act as attention mechanisms

- Suppress less important features maps by give higher weightage to class specifying feature maps
- Can be added to any CNN before convo layer.
- gets global veiw of feature maps

Competitive Squeeze by Excitation 2018:

- In ResNet it only considers the residual information for determining weights of each fe-dure map
- Addressed this problem by generating feature maps-wise motifs (statistic). from both residual by identity mapping based feature maps
- global representation of fM generated using global average pooling

Inner Imaging:
- Relevance between, feature maps estimated by establishing a competition between feature descriptors of residual by identity mapping.

- Not only models the relation between residual feature mapps but also maps their relation with the identity feature maps

Date:

Sun Mon Tue Wed Thu Fri Sat

Channel exploitation based CNN:

- The lack of diversity of class discernable (distinguishing features) information in the input effect CNN's performance

- Introduced channel boosting using auxilary learners.
- exploits channel wise relationships in features maps
- attention mechanisms to adaptively recalibrate feature maps and boost informative content.

Channel Boosted CNN 2018:

- proposed idea of boosting number of input channels for improving the ~~number~~ representational capacity of the network

- artificially create extra channels AKA auxilary channels

$$I_B = g_k(I_c[A_1, \ldots, A_m])$$

$$f_L^K = g_c(I_B, K_L)$$

$I_c \rightarrow$ orignal inputs $A_m \rightarrow$ artificial inputs $g_k \rightarrow$ combiner function

$I_B \rightarrow$ Boosted input F_L^K: feature map generated by convolving I_B with filter K

- for improving representation of data power of TL (transfer learning) by deep generative learner were exploited.

Date:

Sun Mon Tue Wed Thu Fri Sat

- Generative learners learns patterns of input data to create similar data

- Autoencoders are used as generative learners

- Tranfer learners used to create enhaced input representation
  - knowledge gained from one task is used to improve another task

- CB-CNN encodes the channel boosting phase into generic block
- inserted at start of deep Network

- Suggested that boosting block channels can be inserted at any layer in deep architecture.

*

Attention Based CNNs:

- In addition to learning multiple hierarchies of abstractions focusing on features relevant to the context also plays significant role in image localization.

- Views scences in succession of partial glimpse and pay attention to context relevant parts

- RNN & LSTMs exploit attention modules for generation of Sequential data

Date:

Sun Mon Tue Wed Thu Fri Sat

- Idea of attention also help CNN to recognize objects even from cluttered background by complex scenes

- ~~AC~~CNN incorporate attention mechanisms to capture long range dependencies by focus on relevant regions in image

- They also improve network ability to attend to informative parts of the input

Residual Attention Neural Network. 2017:

- proposed to improve feature representation of the network
- ~~Make networks object-aware~~
- Make network capable of learning object-aware feat
- Built by stacking residual blocks with attention module.
- Branched into trunk by mask branches.
  adopts Topdown by bottom up learning

Bottom up: feed forward structure produces low resolution feature maps with strong semantic information

Top down: globally optimizes the network in such a way that it gradually outputs the maps to input & produces dense feature to make infrence of each pixel.

Date:

Sun Mon Tue Wed Thu Fri Sat

$$g_{am}(f_L^k) = g_{sm}(f_L^k) \cdot g_{tm}(f_L^k)$$

g_{sm} → object aware soft mask for feature map at layer L

- assigns attention towards objects

~~g_{tm}~~

- RAN can be made efficient by for cluttered, complex & noisy images by adding attention modules.

- Adaptively assign weight to each feature map based on their relevance in the layer.

- Learning of deep hierarchial structur was supported by residual units

- Mixed, channels, & spatial attentions were incorporated

Convolutional Block Attention 2018:

- SENet only considers the contribution of feature maps in image classification. but ignores the spatial locality of the object in images

- CBAM: infers attention maps sequentially by first applying feature map(channels) attention and then spatial attention to find the refined feature maps

Date:

Sun Mon Tue Wed Thu Fri Sat

- generally 1x1 conv by pooling layer are used for spatial attention.

- CBAM concatenates avg pooling operation with max pooling which generate strong spatial attention map.

- feature map statistics were modelled using combination of max pooling and global ~~max~~ average pooling

- Max pooling could provide clues about distinctive object features

- Global average pooling returns suboptimal inference of feature map attention.

- The exploitation of both pooling improves representational power of network. They not only focus on the important part but also increase the representational power of the selected feature maps

- formulation of 3D attention map via serial learning reduces parameters and computation cost

- CBAM can be inserted in CNN.

Date:

Sun Mon Tue Wed Thu Fri Sat

Concurrent Squeeze & Excitation 2018:

- Extended SENet by incorporating the effect of spatial information in the combination with feature maps (channels) information

- They introduced 3 modules:
  - cSE : Squeezing partially by exciting feature maps information
  - sSE : Squeezing feature map by exciting spatial information
  - scSE : concurrent squeeze by excitation of spatial & fM info

- AE based convolutional NN was used for segmentation
- modules were inserted after the encoders and decoders
-

cSE: • similar concept as SE Block
• Scaling factor is derived based on combination of fM

sSE: • Spatial locality is given more importance
• Different combination of feature maps are selected & exploited spatially to use for segmentation.