African Master's Of Machine Intelligence

AMMI

Densely Connected Convolutional Networks

Engelbert TCHINDE WAMBA
AMMI Student

BootCamp - Week 1

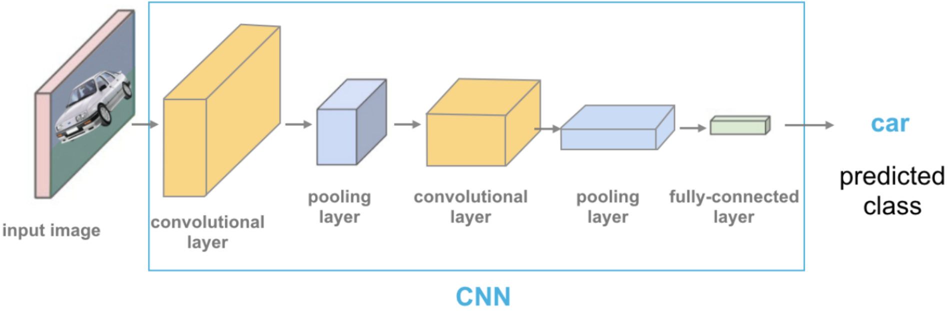
August 20, 2021

Content

- Convolutional Neural Network (CNN)
- DenseNet Architecture
- Multiple densely connected blocks
- Dense connectivity
- DenseNet architecture for ImageNet
- Results
- Conclusion and perspective

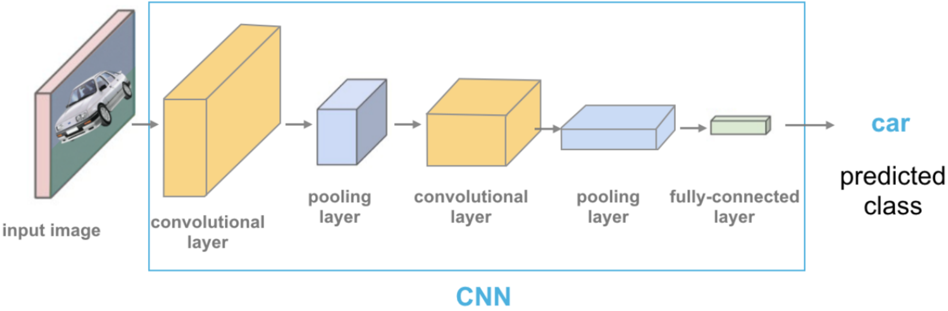
Convolutional Neural Network (CNN)

How do we define CNN?



Convolutional Neural Network (CNN)

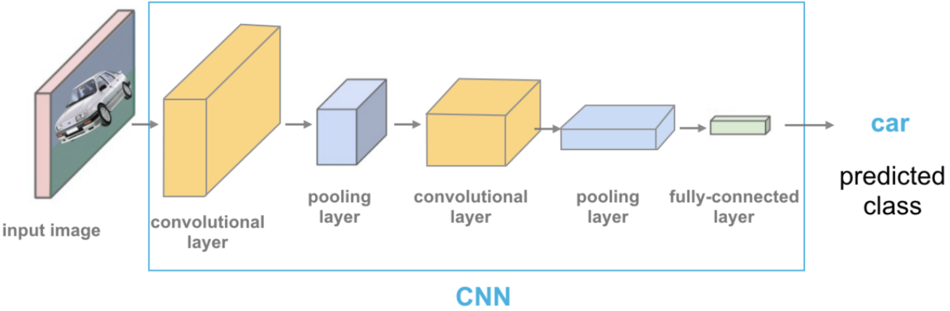
How do we define CNN?



- CNN is a class of deep neural networks, designed for processing structured arrays of data such as images.

Convolutional Neural Network (CNN)

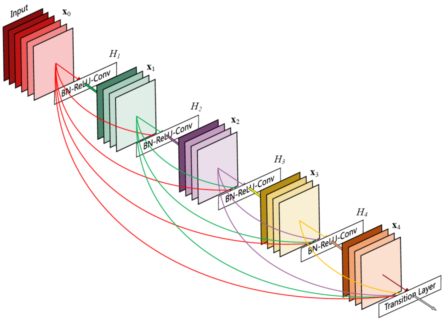
How do we define CNN?



- CNN is a class of deep neural networks, designed for processing structured arrays of data such as images.
- Types: AlexNet - VGGNet - GoogLeNet - ResNet.

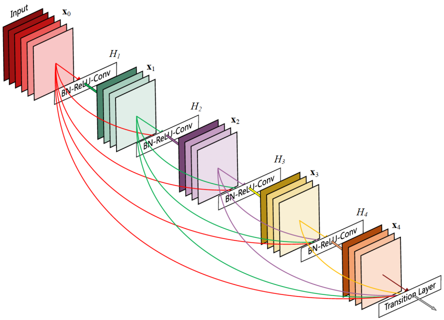
DenseNet Architecture

Why DenseNet?



DenseNet Architecture

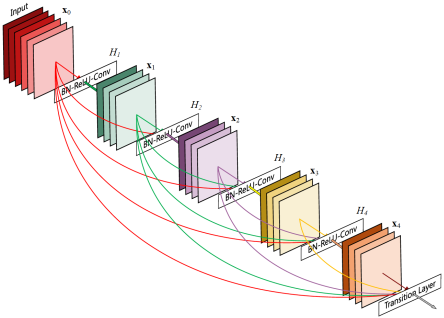
Why DenseNet?



- Every layer is connected to every other layer.

DenseNet Architecture

Why DenseNet?



- Every layer is connected to every other layer.
- The input of a layer inside DenseNet is the concatenation of feature maps from previous layers.

DenseNet Architecture

DenseNets have several compelling advantages.

- they alleviate the vanishing-gradient problem.

DenseNet Architecture

DenseNets have several compelling advantages.

- they alleviate the vanishing-gradient problem.
- strengthen feature propagation.

DenseNet Architecture

DenseNets have several compelling advantages.

- they alleviate the vanishing-gradient problem.
- strengthen feature propagation.
- encourage feature reuse.

DenseNet Architecture

DenseNets have several compelling advantages.

- they alleviate the vanishing-gradient problem.
- strengthen feature propagation.
- encourage feature reuse.
- substantially reduce the number of parameters.

Multiple densely connected blocks

- Feature maps the same size problem throughout the network.

Multiple densely connected blocks

- Feature maps the same size problem throughout the network.

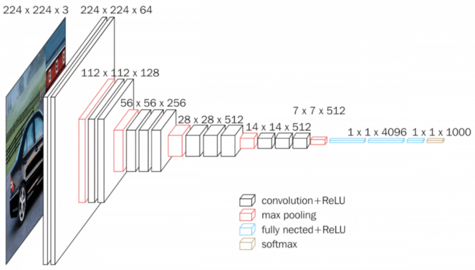


Figure: VGG architecture.

Multiple densely connected blocks

- Feature maps the same size problem throughout the network.

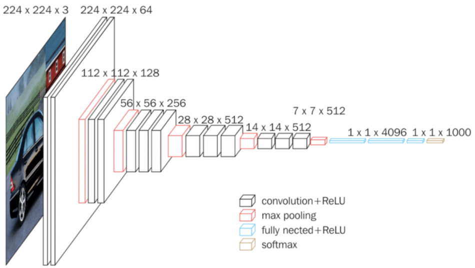
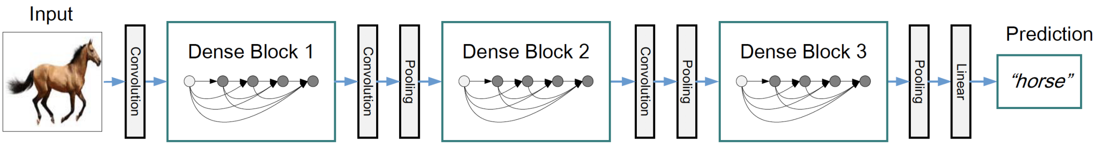


Figure: VGG architecture.

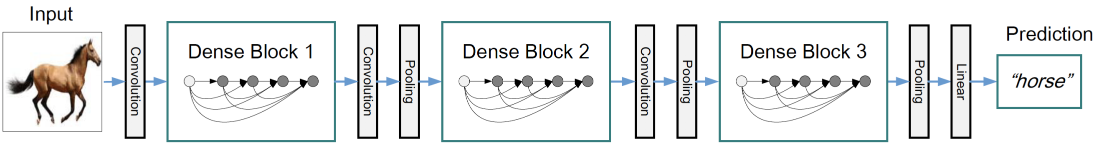
- Essential part of concvolutional networks: down-sampling layers that change the size of feature maps.

Multiple densely connected blocks



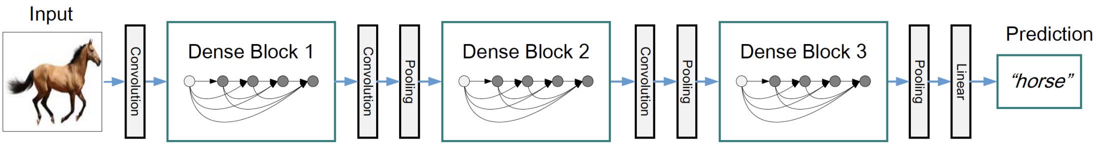
- Solution: divided the network into multiple densely connected dense blocks.

Multiple densely connected blocks



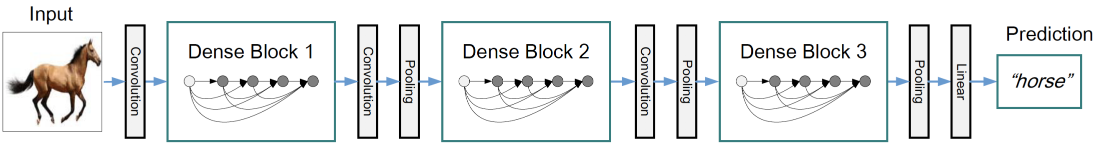
- Solution: divided the network into multiple densely connected dense blocks.
- Inside the dense blocks the feature map size remains the same.

Multiple densely connected blocks



- Solution: divided the network into multiple densely connected dense blocks.
- Inside the dense blocks the feature map size remains the same.
- The size of the feature map grows after a pass through each dense layer and the new features are concatenated to the existing features.

Multiple densely connected blocks



- Solution: divided the network into multiple densely connected dense blocks.
- Inside the dense blocks the feature map size remains the same.
- The size of the feature map grows after a pass through each dense layer and the new features are concatenated to the existing features.
- Each layer adds k features on top to the global state, k growth rate of the network.

Dense connectivity

- Traditional convolutional feed-forward networks connect the output of the ℓ^{th} layer to the $(\ell + 1)^{th}$ layer. Layer transition: $\mathbf{x}_\ell = \mathbf{H}_\ell(\mathbf{x}_{\ell-1})$.

Dense connectivity

- Traditional convolutional feed-forward networks connect the output of the ℓ^{th} layer to the $(\ell+1)^{th}$ layer. Layer transition: $\mathbf{x}_\ell = \mathbf{H}_\ell(\mathbf{x}_{\ell-1})$.
- The skip connection can be represented as:

$$\mathbf{x}_\ell = \mathbf{H}_\ell(\mathbf{x}_{\ell-1}) + \mathbf{x}_{\ell-1}. \tag{1}$$

Dense connectivity

- Traditional convolutional feed-forward networks connect the output of the ℓ^{th} layer to the $(\ell+1)^{th}$ layer. Layer transition: $\mathbf{x}_\ell = \mathbf{H}_\ell(\mathbf{x}_{\ell-1})$.
- The skip connection can be represented as:

$$\mathbf{x}_\ell = \mathbf{H}_\ell(\mathbf{x}_{\ell-1}) + \mathbf{x}_{\ell-1}. \tag{1}$$

- In DenseNet architecture, the dense connectivity can be represented as:

$$\mathbf{x}_\ell = \mathbf{H}_\ell([\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_{\ell-1}]) \tag{2}$$

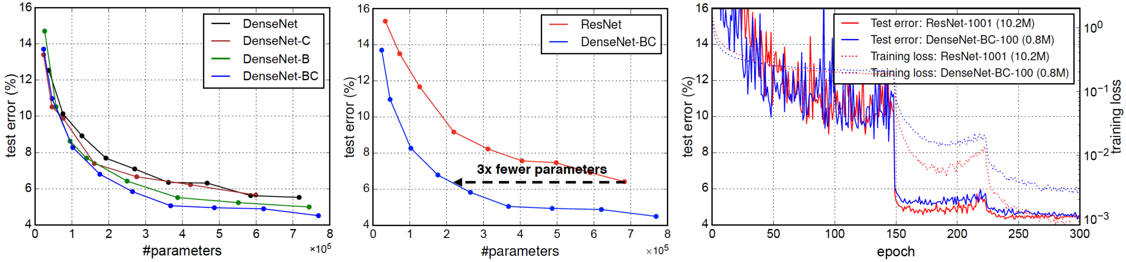
DenseNet architectures for ImageNet

Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Result: Error rate

Method	Depth	Params	C10	C10+	C100	C100+	SVHN
Network in Network [22]	-	-	10.41	8.81	35.68	-	2.35
All-CNN [32]	-	-	9.08	7.25	-	33.71	-
Deeply Supervised Net [20]	-	-	9.69	7.97	-	34.57	1.92
Highway Network [34]	-	-	-	7.72	-	32.39	-
FractalNet [17]	21	38.6M	10.18	5.22	35.34	23.30	2.01
with Dropout/Drop-path	21	38.6M	7.33	4.60	28.20	23.73	1.87
ResNet [11]	110	1.7M	-	6.61	-	-	-
ResNet (reported by [13])	110	1.7M	13.63	6.41	44.74	27.22	2.01
ResNet with Stochastic Depth [13]	110	1.7M	11.66	5.23	37.80	24.58	1.75
	1202	10.2M	-	4.91	-	-	-
Wide ResNet [42]	16	11.0M	-	4.81	-	22.07	-
	28	36.5M	-	4.17	-	20.50	-
with Dropout	16	2.7M	-	-	-	-	1.64
ResNet (pre-activation) [12]	164	1.7M	11.26*	5.46	35.58*	24.33	-
	1001	10.2M	10.56*	4.62	33.47*	22.71	-
DenseNet ($k = 12$)	40	1.0M	**7.00**	5.24	**27.55**	24.42	1.79
DenseNet ($k = 12$)	100	7.0M	**5.77**	**4.10**	**23.79**	**20.20**	1.67
DenseNet ($k = 24$)	100	27.2M	**5.83**	**3.74**	**23.42**	**19.25**	**1.59**
DenseNet-BC ($k = 12$)	100	0.8M	**5.92**	4.51	**24.15**	22.27	1.76
DenseNet-BC ($k = 24$)	250	15.3M	**5.19**	**3.62**	**19.64**	**17.60**	1.74
DenseNet-BC ($k = 40$)	190	25.6M	-	**3.46**	-	**17.18**	-

Result: Comparison of the parameter



Conclusion and perspective

- DenseNets achieved state-of-the-art results on several highly competitive datasets.

Conclusion and perspective

- DenseNets achieved state-of-the-art results on several highly competitive datasets.
- They require far fewer parameters and less computation to achieve the best performance.

Conclusion and perspective

- DenseNets achieved state-of-the-art results on several highly competitive datasets.
- They require far fewer parameters and less computation to achieve the best performance.
- Future work: study such feature transfer with DenseNets

Reference

Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger; *Densely Connected Convolutional Networks*. CVPR 2017.

End

Thank you for your attention!