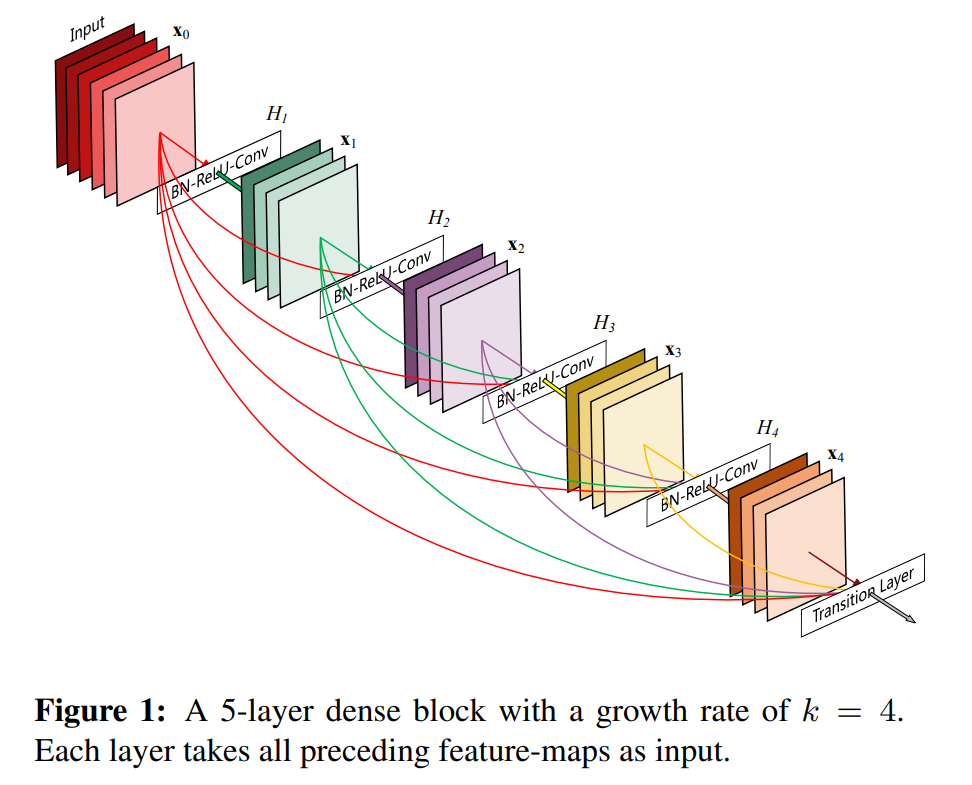
**Densely Connected Convolutional Networks**

*Gao Huang, Zhuang Liu, et al.*

* Source: <https://arxiv.org/pdf/1608.06993.pdf>
* Traditional convolutional networks with L layers have L connections – one between each layer and its subsequent layer – our network has direct connections.
* Each layer receives outputs of all the previous layers as input.
* DenseNet Advantages:
  + alleviate vanishing-gradient
  + strengthen feature propagation
  + encourage feature reuse
  + reduce the number of parameters
* Like ResNet, this network aims to pass information from some early layer to some other layer deeper in the network. This helps in dealing with vanishing information or gradient.



* In contrast to ResNets, we never combine features through summation before they are passed into a layer; instead, we combine features by concatenating them. Hence, the *lth* layer has *l* inputs, consisting of the feature-maps of all preceding convolutional blocks.
* Because of the dense connectivity pattern, it is called DenseNet.
* A possibly counter-intuitive effect of this dense connectivity pattern is that it requires fewer parameters than traditional convolutional networks, as there is no need to relearn redundant feature-maps.
* Tradition CNN can be thought of as each layer learning a state and then passing it to the next state. The state that a layer outputs will have information from its input state and some new information it learnt.
* DenseNet basically separates this. The states output by all the previous layers will be presented to a layer all at once (by concatenation along channels).

So, each layer doesn’t need to learn a *modified* state that has input state and some new information. Instead, each layer just outputs state that has the new information.

Thus, DenseNet layers are very narrow (e.g., 12 filters per layer)

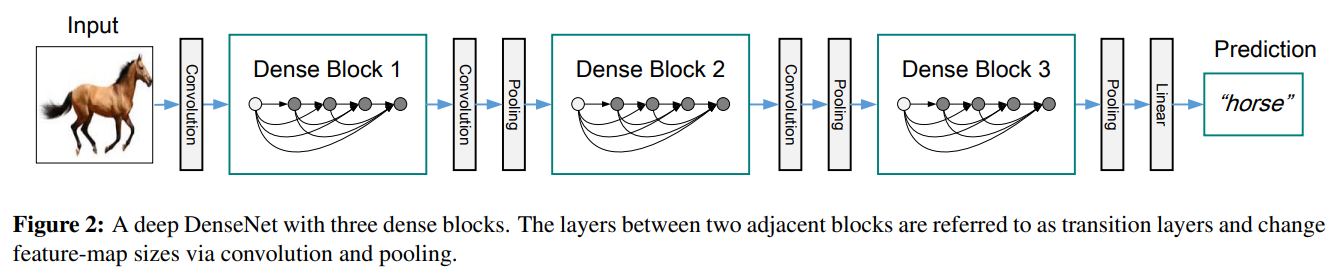
* This achieves better parameter efficiency. Also, it improves flow of information and gradients throughout the network, which makes training easier.

Each layer receives gradient directly from the loss function.

* Dense connections have a regularizing effect, which reduces overfitting on tasks with smaller training set sizes.
* Channel-wise concatenation of the outputs of all the previous layers is possible only if they have same spatial dimensions. However, a network also needs to reduce the dimensions as we go deeper into the network.

So, DenseNet has multiple *dense blocks*. In a dense block, the input to a layer is the concatenation of the outputs of all the previous layers **within the same dense block.**

* Downsampling happens between dense blocks. These downsampling layers are called *transition layers*. In DenseNet, transition layers include Batch-norm layer, 1\*1 conv layer, and then 2\*2 average pooling layer.



* If each function produces k feature maps, it follows that the *lth* layer has *k0 + k \* (l−1)* input feature-maps, where *k0* is the number of channels in the input layer.

Basically, each layer in a dense block produces output having K channels; in other words, all layers have K filters.

DenseNet has k=12. This parameter is called *Growth Rate*.

* The feature maps can be viewed as the *global state* of the network. Each layer adds k feature maps to this global state. The growth rate basically controls how much new information each layer contributes to the global state.
* While each layer outputs only k feature maps, the feature map size input to the layers increase as we go deeper into the network. For example, the first layer will have k0 feature maps, the second layer will have k0+k feature maps, the third layer will have k0+2k feature maps, and so on.

To reduce this, use 1\*1 conv layer as bottleneck layer before each 3\*3 conv layer.

So, the structure becomes *BN-ReLU-Conv(1×1)-BN-ReLU-Conv(3×3)*. Such network is called DenseNet-B. (B for bottleneck)

In DenseNet-B, the bottleneck layer reduces the input feature maps to 4k.

* The feature maps can be reduced further by transition layers. If the input to a transition layer has feature maps, the output has feature maps, where

Such network with is called DenseNet-C. (C for compression). Authors set

A DenseNet with bottleneck and compression design is called DenseNet-BC.

* Training

SKIPPED – not important

* Results:

SKIPPED – not important