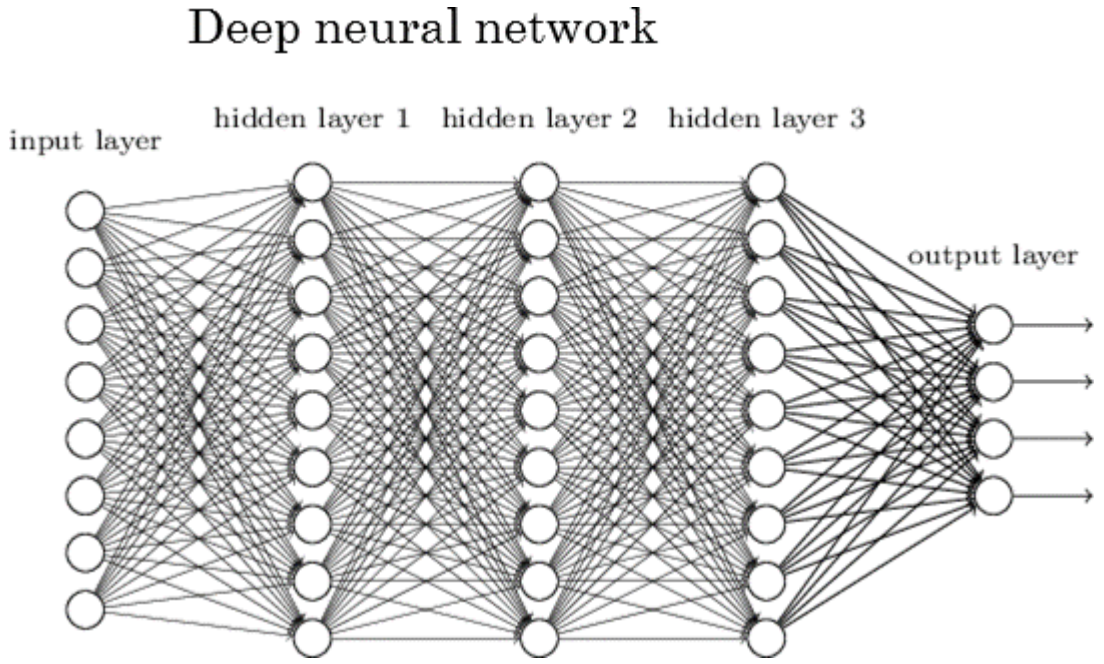
The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation

Simon Jegou , Michal Drozdzal, David Vazquez, Adriana Romero
Yoshua Bengio

Deep Neural Network

- use a cascade of multiple layers of units for feature extraction. Each successive layer uses the output from the previous layer as input.



Deep Neural Network

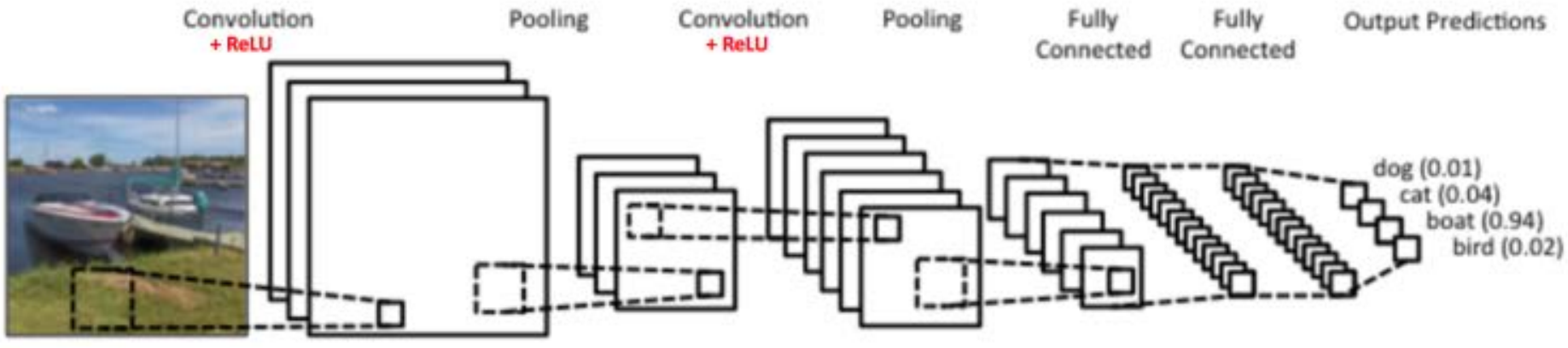
- Regular Neural Nets don’t scale well to full images

- 32*32*3 (32 wide, 32 high, 3 color channels), so a single fully-connected neuron in a first hidden layer of a regular Neural Network would have 32*32*3 = 3072 weights.

- we would almost certainly want to have several such neurons, so the parameters would add up quickly! Clearly, this full connectivity is wasteful and the huge number of parameters would quickly lead to overfitting.

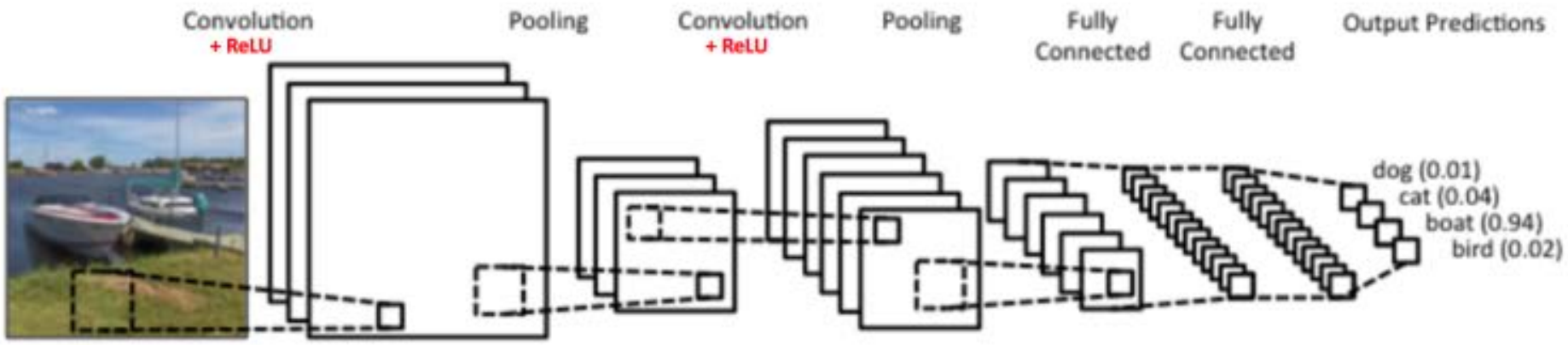
Convolutional Neural Network

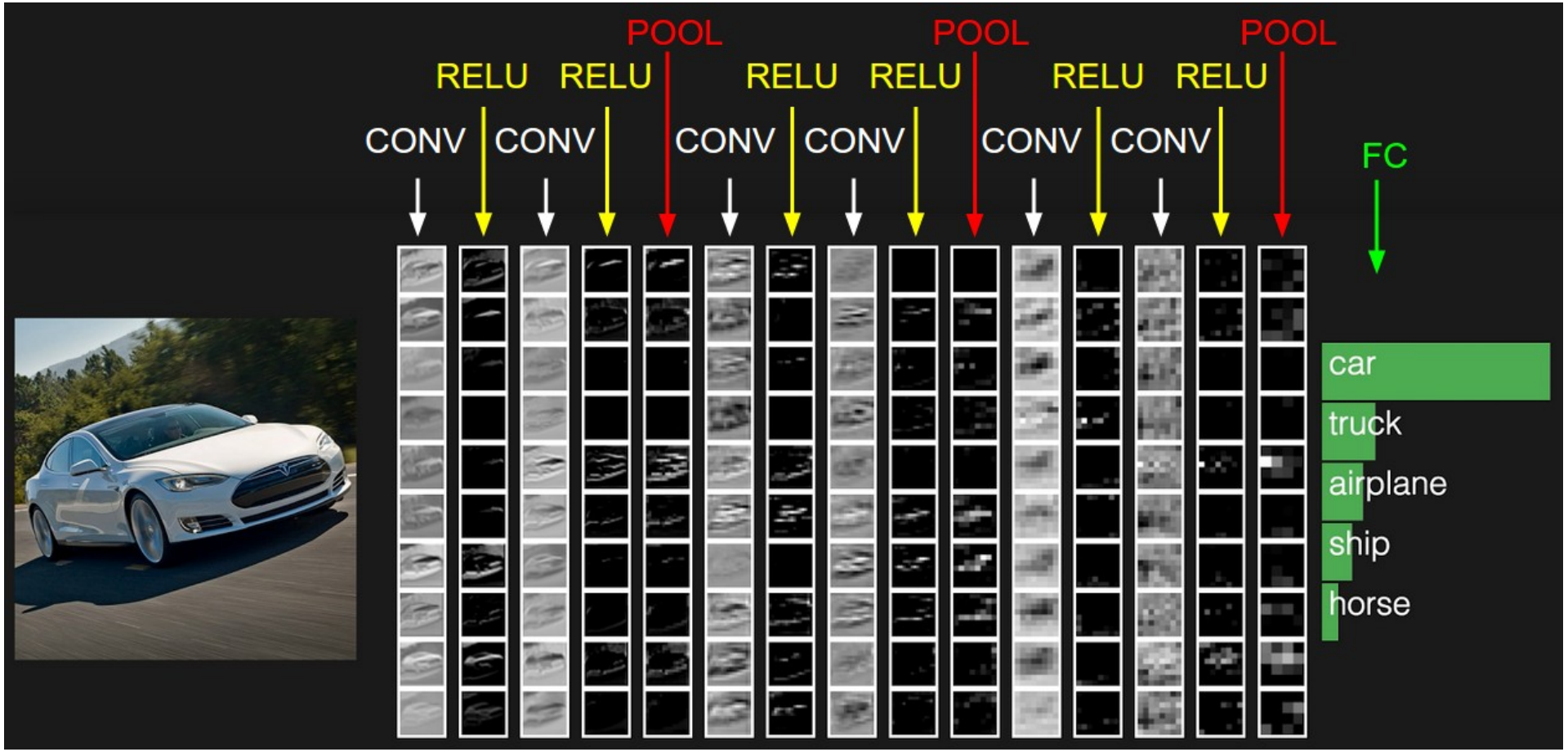
- **convolutional neural network** (**CNN**, or **ConvNet**) is a class of deep artificial neural network that has successfully been applied to analyzing visual imagery.



Convolutional Neural Network

- connect each neuron to only a local region of the input volume. The spatial extent of this connectivity is a hyperparameter called the **receptive field** of the neuron (equivalently this is the filter size).

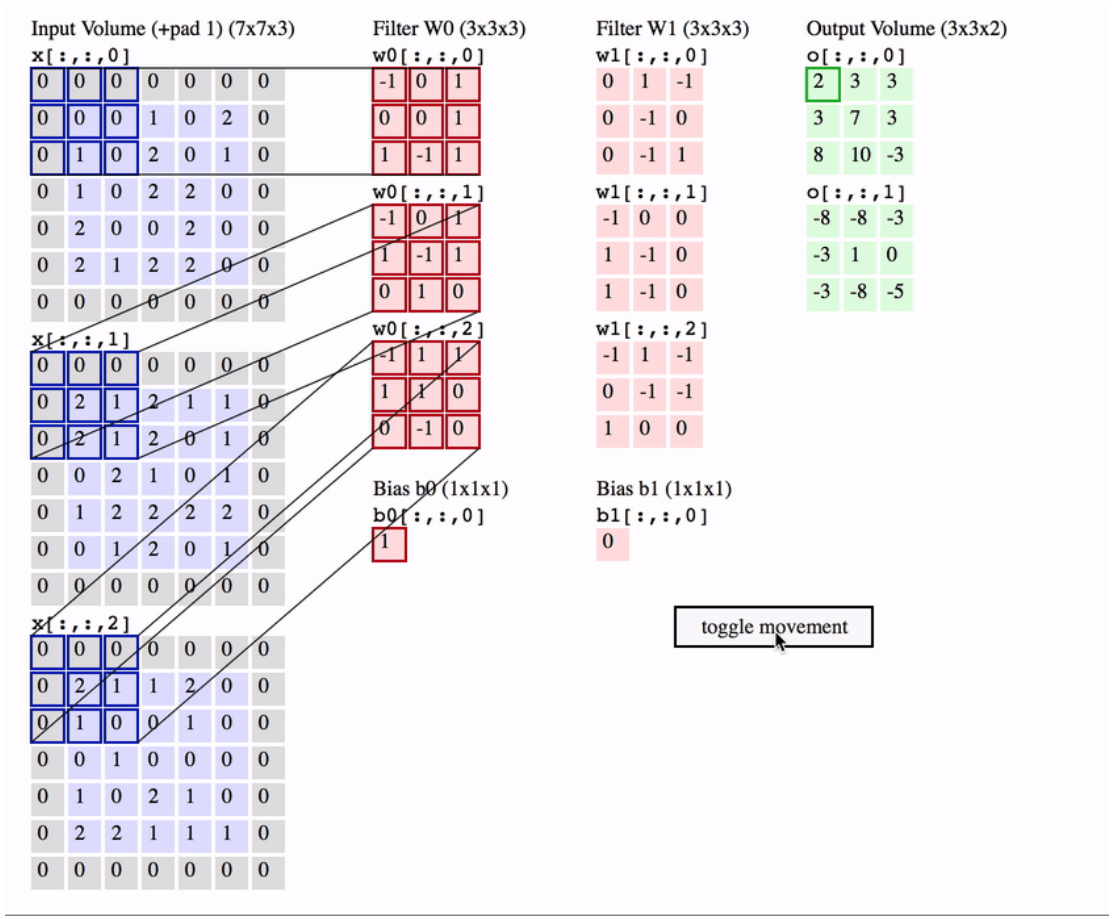



POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

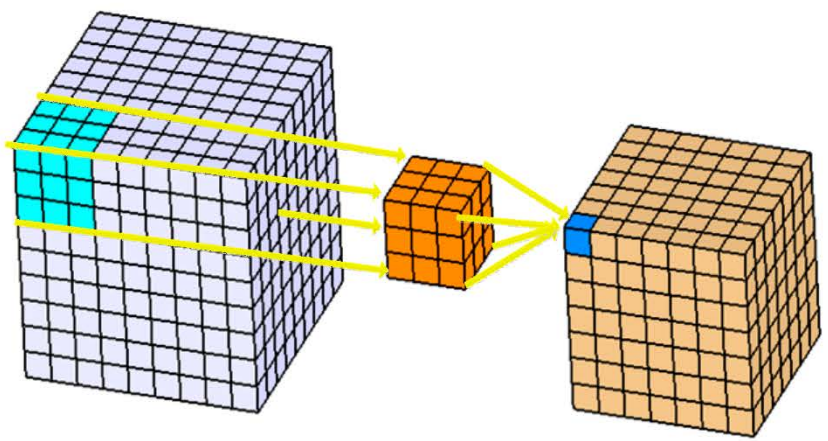
Convolution

- Convolutional layers apply a convolution operation to the input, passing the result to the next layer.
- Each convolutional neuron processes data only for its receptive field.
- http://cs231n.github.io/convolutional-networks/
- https://github.com/vdumoulin/conv_arithmetic

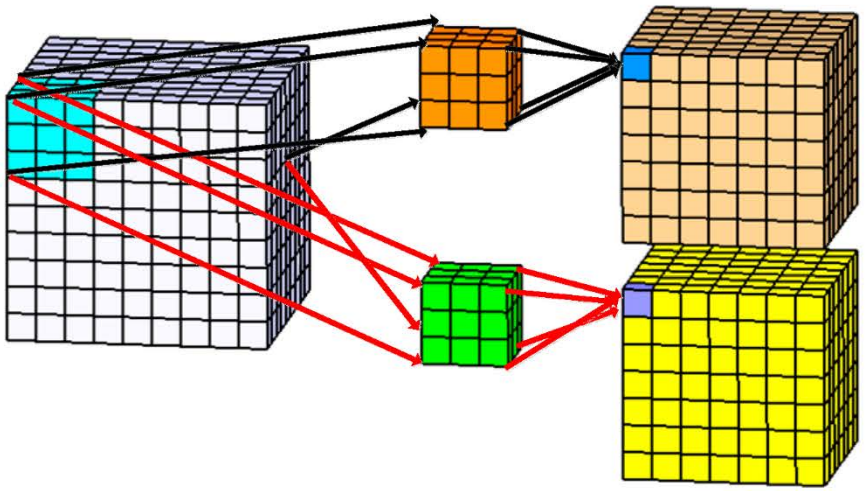
Convolution



3D convolution



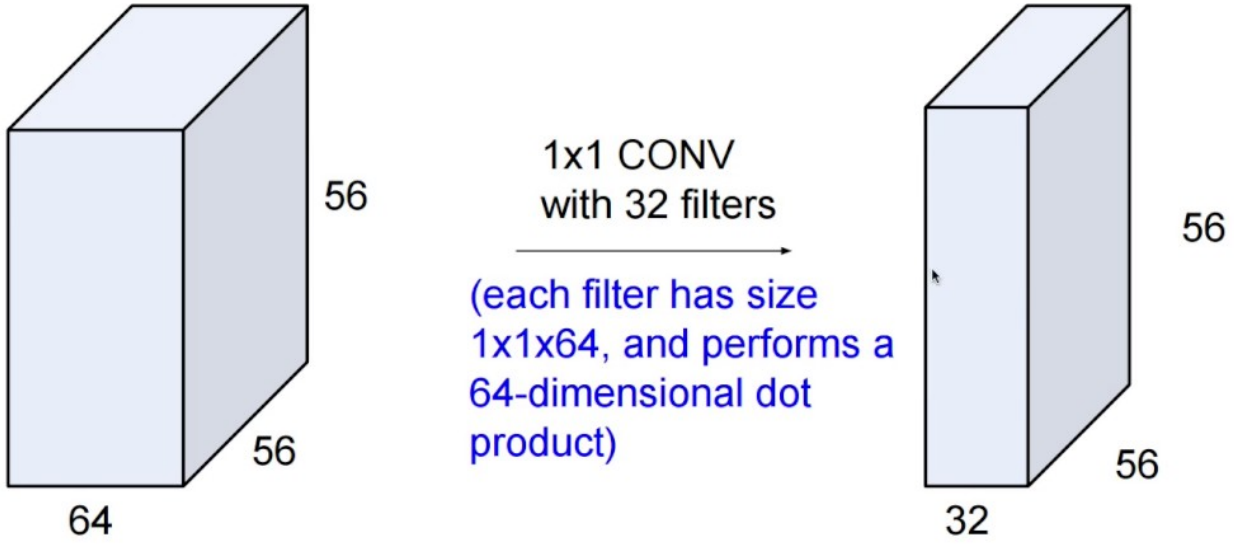
(a)



(b)

1*1 Convolution

(btw, 1x1 convolution layers make perfect sense)



Rectifier (ReLU)

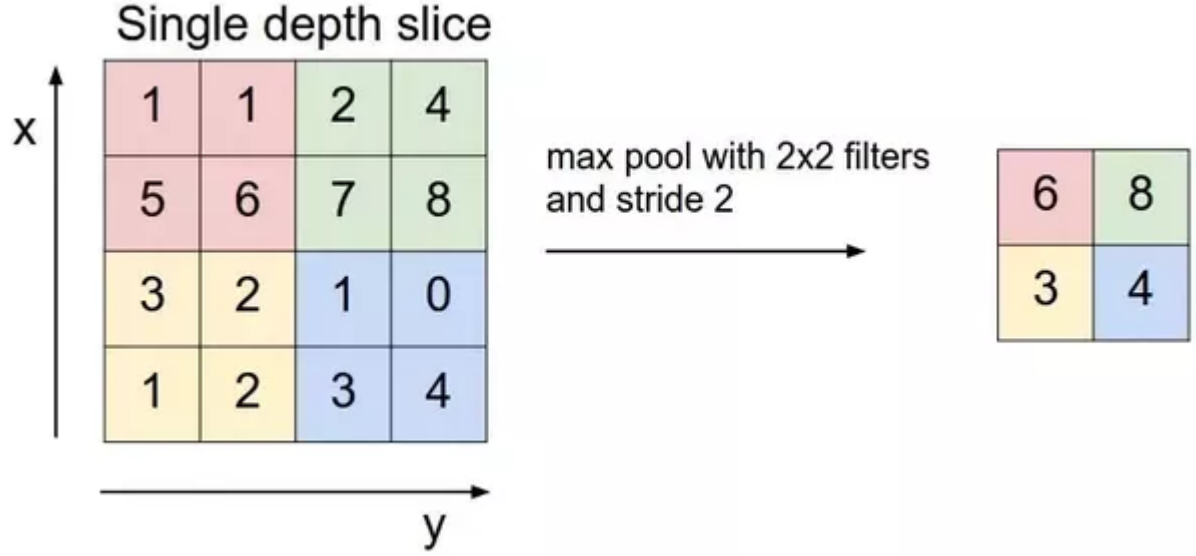
- the **rectifier** is an activation function defined as the positive part of its argument:

$$f(x) = x^+ = \max(0, x),$$

- where *x* is the input to a neuron.

Pooling

- Pooling is a **sample-based discretization process**. The objective is to down-sample an input representation.
- Max pooling
- Average pooling



Batch Normalization

- Batch Normalization is a method to reduce internal covariate shift in neural networks.

- We define Internal Covariate Shift as the change in the distribution of network activations due to the change in network parameters during training.
- https://machinelearning.wtf/terms/internal-covariate-shift/
- https://wiki.tum.de/display/lfdv/Batch+Normalization

Dropout

- **Dropout** is a regularization technique for educing over fitting in neural network by preventing complex co-adaptations on training data.
- The term "dropout" refers to dropping out units.

Transpose convolution (De convolution)

https://github.com/vdumoulin/conv_arithmetic

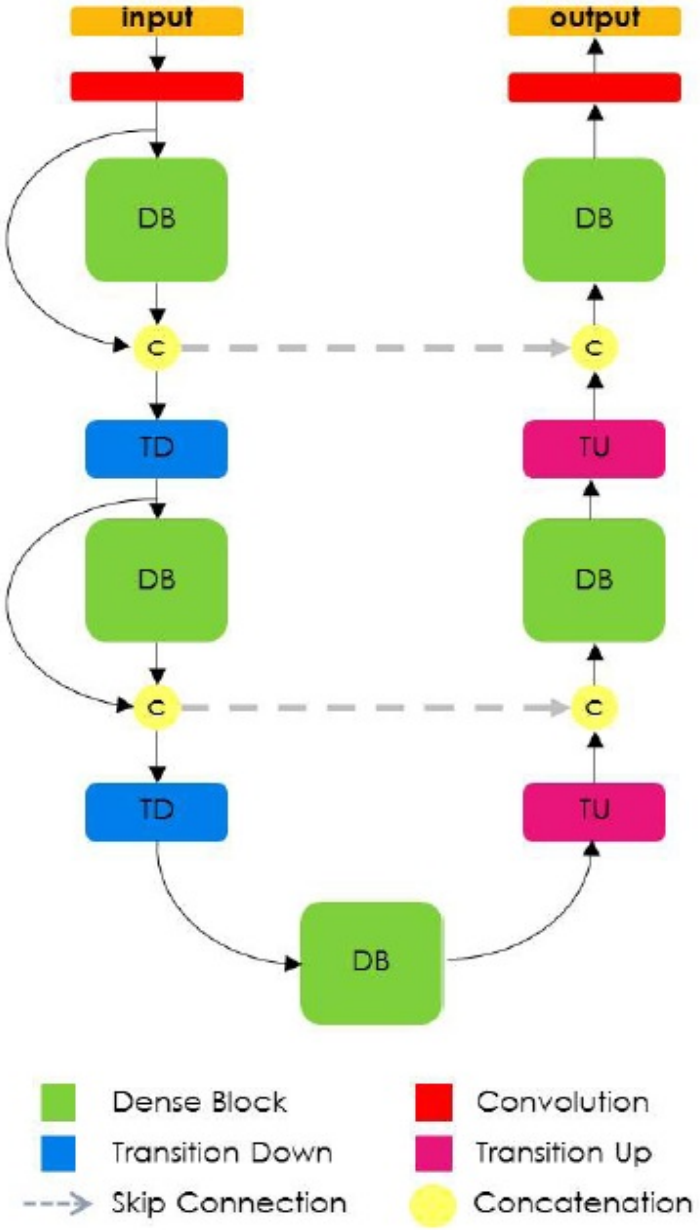
https://www.quora.com/What-is-the-difference-between-Deconvolution-Upsampling-Unpooling-and-Convolutional-Sparse-Coding

Abstract

The typical segmentation architecture is composed of :

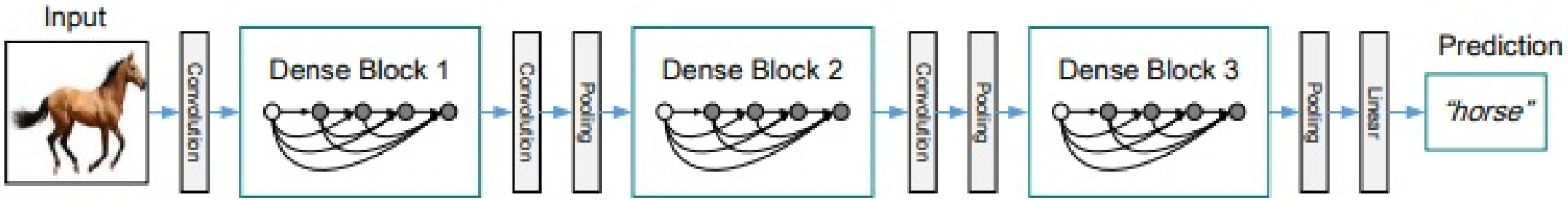
- a downsampling path responsible for extracting coarse semantic features.
- an upsampling path trained to recover the input image resolution at the output of the model.
- optionally, a post-processing module.

Abstract


input
output
DB
DB
C
C
TD
TU
DB
DB
C
C
TD
TU
DB
Dense Block
Convolution
Transition Down
Transition Up
Skip Connection
Concatenation

Abstract

- Densely Connected Convolutional Networks (DenseNets)
- https://arxiv.org/abs/1608.06993



Abstract

achieve state-of-the-art results on urban scene benchmark datasets:

- CamVid is a dataset of fully segmented videos for urban scene understanding.
- http://mi.eng.cam.ac.uk/research/projects/VideoRec/CamVid/
- Gatech is a geometric scene understanding dataset.
- http://www.cc.gatech.edu/cpl/projects/videogeometriccontext/

Introduction

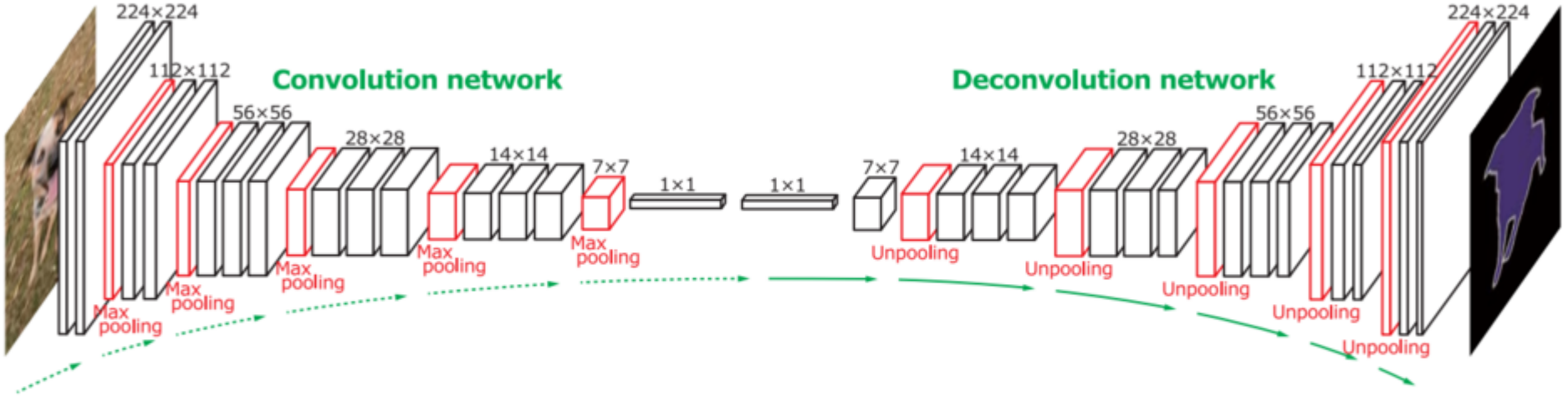


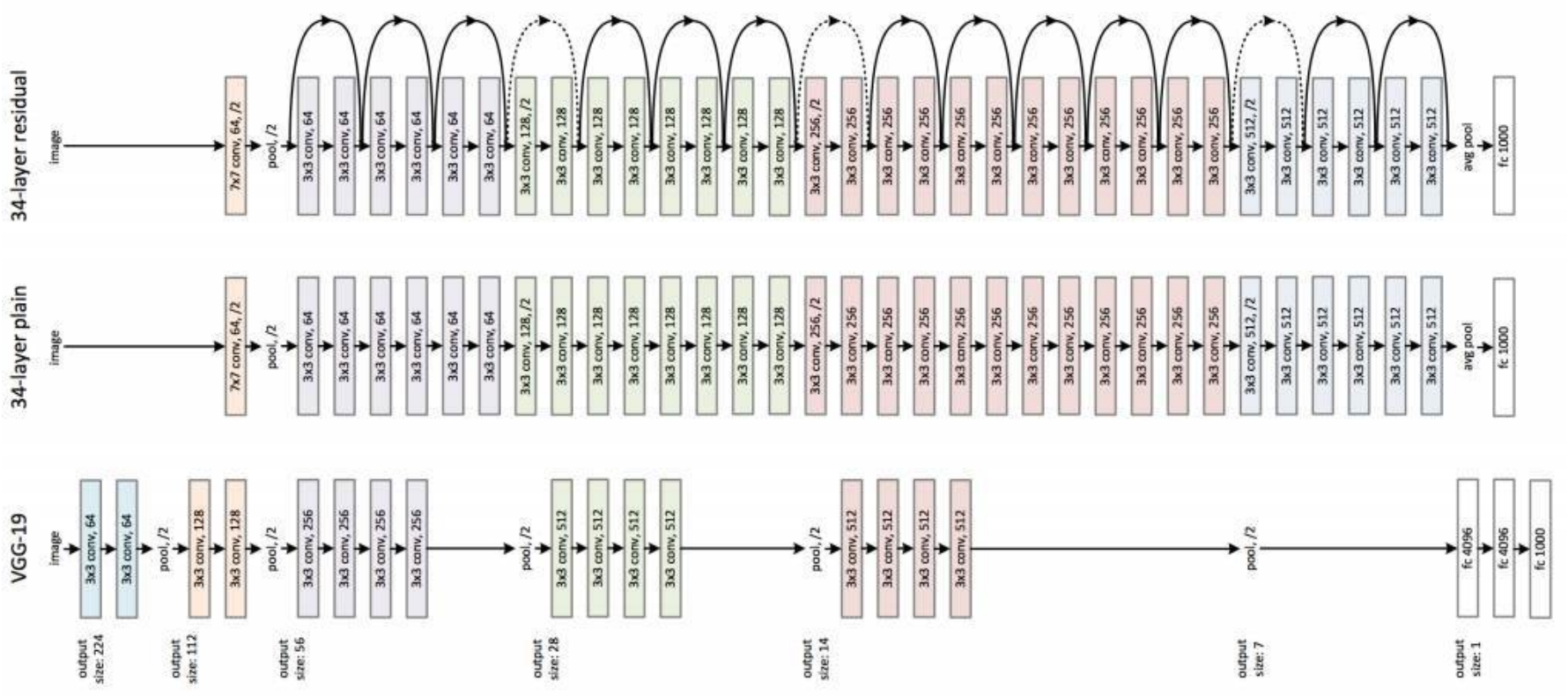
Figure 2. Overall architecture of the proposed network. On top of the convolution network based on VGG 16-layer net, we put a multi-layer deconvolution network to generate the accurate segmentation map of an input proposal. Given a feature representation obtained from the convolution network, dense pixel-wise class prediction map is constructed through multiple series of unpooling, deconvolution and rectification operations.

Introduction

- Res net
- https://arxiv.org/abs/1512.03385
- Unet
- https://arxiv.org/abs/1505.04597

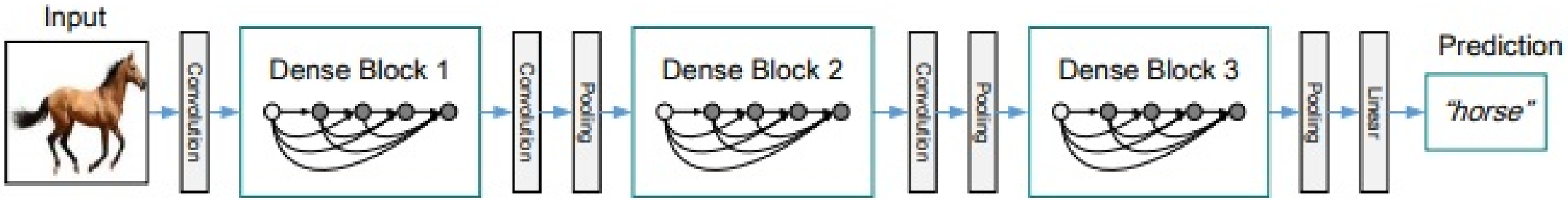
Introduction

ResNet



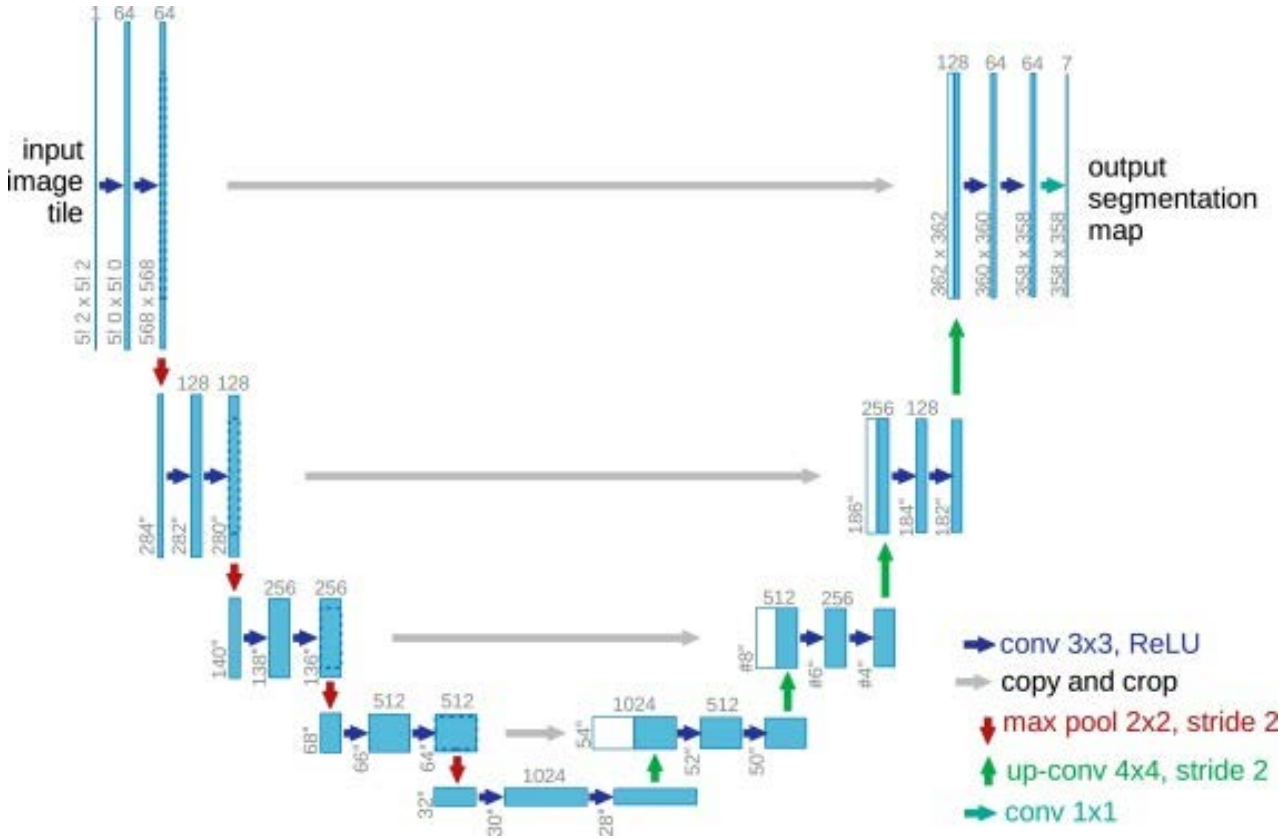
Review of DensNet

- Densely Connected Convolutional Networks (DenseNets)
- https://arxiv.org/abs/1608.06993

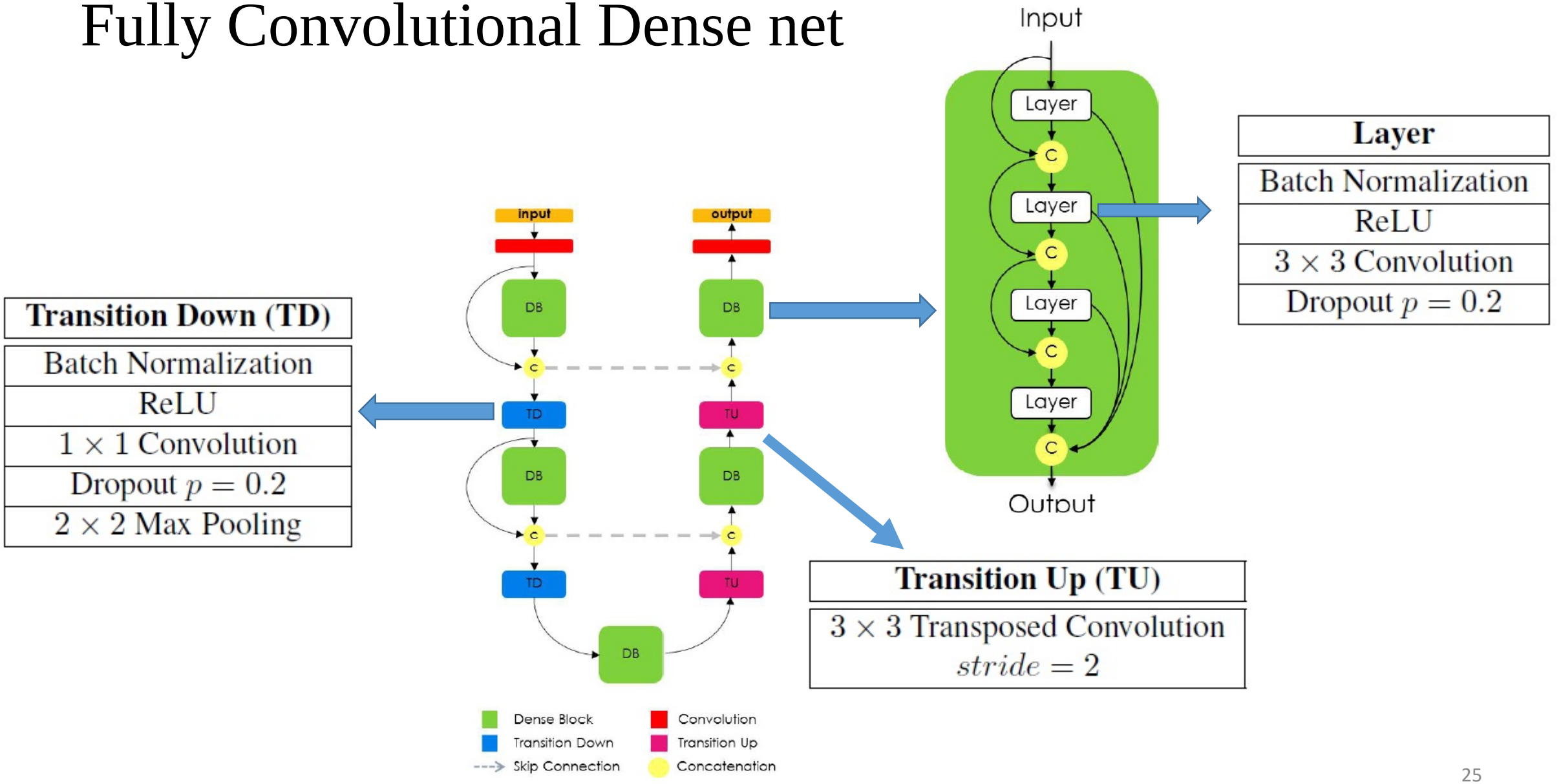


Introduction

UNet



Fully Convolutional Dense net



Layer
Batch Normalization
ReLU
3×3 Convolution
Dropout $p = 0.2$

Transition Down (TD)
Batch Normalization
ReLU
1×1 Convolution
Dropout $p = 0.2$
2×2 Max Pooling

Transition Up (TU)
3×3 Transposed Convolution $stride = 2$

Architecture
Input, $m = 3$
3×3 Convolution, $m = 48$
DB (4 layers) + TD, $m = 112$
DB (5 layers) + TD, $m = 192$
DB (7 layers) + TD, $m = 304$
DB (10 layers) + TD, $m = 464$
DB (12 layers) + TD, $m = 656$
DB (15 layers), $m = 896$
TU + DB (12 layers), $m = 1088$
TU + DB (10 layers), $m = 816$
TU + DB (7 layers), $m = 578$
TU + DB (5 layers), $m = 384$
TU + DB (4 layers), $m = 256$
1×1 Convolution, $m = c$
Softmax

Soft max

$$\varphi(\mathbf{x})_j = \frac{e^{\mathbf{x}_j}}{\sum_{k=1}^{K} e^{\mathbf{x}_k}}$$

Parameters:	**x** : float32 The activation (the summed, weighted input of a neuron).
Returns:	float32 where the sum of the row is 1 and each single value is in [0, 1] The output of the softmax function applied to the activation.

Heuniform

- https://arxiv.org/abs/1502.01852
- This leads to a zero-mean Gaussian distribution whose standard deviation (std) is $\sqrt{2/n_l}$.
- We use *l* to index a layer and *n* denoting number of layer connections.

RMSprop

- http://ruder.io/optimizing-gradient-descent/index.html#rmsprop