



UPPSALA
UNIVERSITET

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Machine learning – Block 4

Måns Magnusson
Department of Statistics, Uppsala University

Autumn 2023



- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
-
- Previous assignments
 - The Mini-Project and Master Thesis Projects
 - Convolutional Neural Networks
 - Transfer Learning



Assignment 3: Evaluation

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

- TODO



On this weeks assignment

- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- It takes long time to run the models this week. Start early!



Mini-project

- Previous assignments
 - **The Mini-Project**
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- 
- A faint, decorative watermark of a crown or crest is visible at the bottom of the slide.



Mini-project

- Previous assignments
- **The Mini-Project**
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

- Submission Wednesday!
- Supervised problem of choice on **real data**.



- Previous assignments
- **The Mini-Project**
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Mini-project

- Submission Wednesday!
- Supervised problem of choice on **real data**.
- 2-3 students.



- Previous assignments
- **The Mini-Project**
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Mini-project

- Submission Wednesday!
- Supervised problem of choice on **real data**.
- 2-3 students.
- *Hint!* Submit page 1 of the project as project proposal.
- Feel free to combine it with your master thesis project!
- Check with me if you have questions.
- The project should result in a 4 page report (PDF) using the **ICML LaTeX template**.



UPPSALA
UNIVERSITET

- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Section 3

Introduction



Convolutional Neural Networks

- Acknowledgements: Anders Eklund, Linköping University.

- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology





- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolutional Neural Networks

- **Acknowledgements:** Anders Eklund, Linköping University.
- **Convolutional** Neural Networks are behind great progress in the 2010s.
- Revolutionized **Computer Vision**.

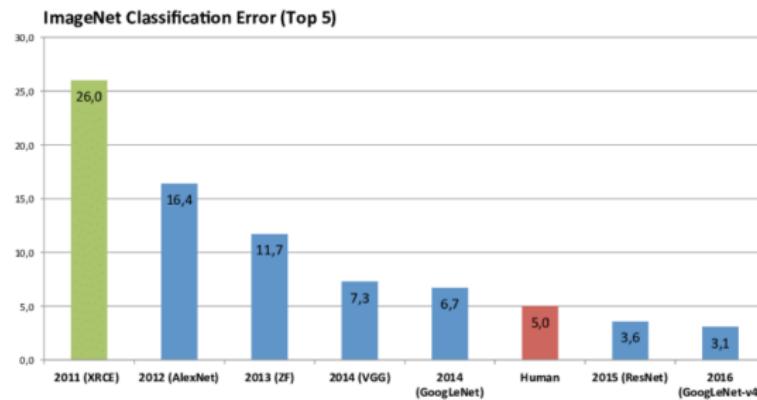


- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolutional Neural Networks

- **Acknowledgements:** Anders Eklund, Linköping University.
- **Convolutional** Neural Networks are behind great progress in the 2010s.
- Revolutionized **Computer Vision**.
- Also called: ConvNets, Convolutional nets, Convolutional networks

Figure: ImageNet performance (Roessler, 2019)



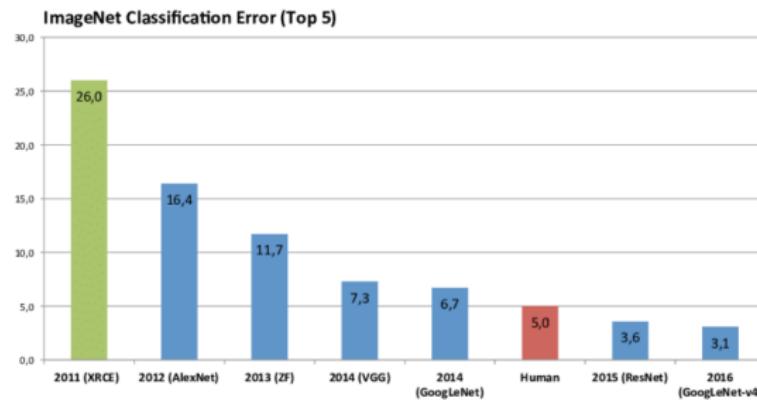


- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolutional Neural Networks

- **Acknowledgements:** Anders Eklund, Linköping University.
- **Convolutional** Neural Networks are behind great progress in the 2010s.
- Revolutionized **Computer Vision**.
- Also called: ConvNets, Convolutional nets, Convolutional networks

Figure: ImageNet performance (Roessler, 2019)





- Previous assignments
 - The Mini-Project
 - **Introduction**
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- Special architecture that works well for data with a **grid structure**
 1. 1D-grids: Time series



- Previous assignments
 - The Mini-Project
 - **Introduction**
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- Special architecture that works well for data with a **grid structure**
 1. 1D-grids: Time series
 2. 2D-grids: Gray-scale Images (pixels)



- Previous assignments
 - The Mini-Project
 - **Introduction**
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- Special architecture that works well for data with a **grid structure**
 1. 1D-grids: Time series
 2. 2D-grids: Gray-scale Images (pixels)
 3. 3D-grids: Color Images (pixels and channels)



- Previous assignments
 - The Mini-Project
 - **Introduction**
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- Special architecture that works well for data with a **grid structure**
 1. 1D-grids: Time series
 2. 2D-grids: Gray-scale Images (pixels)
 3. 3D-grids: Color Images (pixels and channels)
 4. 4D-grids: Color Video (pixels, channels, frames)



- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

- Problems
 - Image Classification



- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

- Problems
 - Image Classification
 - Image Segmentation



- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Computer Vision

- Problems
 - Image Classification
 - Image Segmentation
 - Object Detection



- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

● Problems

- Image Classification
- Image Segmentation
- Object Detection
- Object Localization



- Previous assignments
 - The Mini-Project
 - **Introduction**
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- **Problems**
 - Image Classification
 - Image Segmentation
 - Object Detection
 - Object Localization
 - **Focus:** 2D and 3D data





- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Computer Vision

- **Problems**
 - Image Classification
 - Image Segmentation
 - Object Detection
 - Object Localization
- **Focus:** 2D and 3D data
- **Very Large Datasets:**
 - ImageNet: 14M Images, 20k classes, 1M bounding boxes



- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Computer Vision

- **Problems**
 - Image Classification
 - Image Segmentation
 - Object Detection
 - Object Localization
- **Focus:** 2D and 3D data
- **Very Large Datasets:**
 - ImageNet: 14M Images, 20k classes, 1M bounding boxes
 - Many different pre-trained models (e.g. VGG16)



- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology



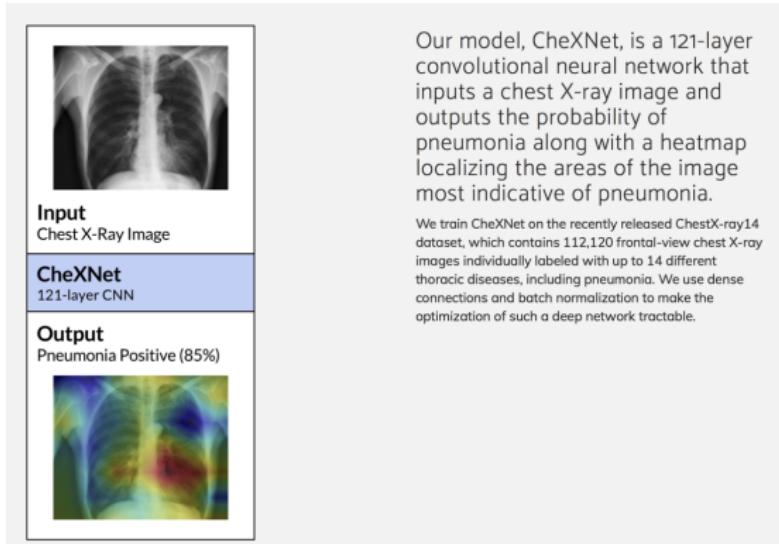
Figure: Object detection (see
<https://www.youtube.com/watch?v=VOC3huqHrss>)



UPPSALA UNIVERSITET

- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Example: Pneumonia detection



Our model, CheXNet, is a 121-layer convolutional neural network that inputs a chest X-ray image and outputs the probability of pneumonia along with a heatmap localizing the areas of the image most indicative of pneumonia.

We train CheXNet on the recently released ChestX-ray14 dataset, which contains 112,120 frontal-view chest X-ray images individually labeled with up to 14 different thoracic diseases, including pneumonia. We use dense connections and batch normalization to make the optimization of such a deep network tractable.

Figure: Rajpurkar et al. (2017). Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225.



- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Example: Fracture detection



Figure 1. 2 images from the dataset. The area within the red box is the section presented to the network in order to classify the image. The left image is of a wrist fracture while the right image is without any apparent fracture.

Figure: Olczak et al, (2017) Artificial intelligence for analyzing orthopedic trauma radiographs, Acta Orthopaedica, 88:6, 581-586



UPPSALA
UNIVERSITET

- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

What is an Image?

- 2-dimensional object
- Each pixel has:
 1. a coordinate
 2. a value (light intensity)



- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

What is an Image?

- 2-dimensional object
- Each pixel has:
 1. a coordinate
 2. a value (light intensity)
- **Grayscale:** single channel
- **Color:** three channel (RGB)



- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

What is an Image?

- 2-dimensional object
- Each pixel has:
 1. a coordinate
 2. a value (light intensity)
- **Grayscale:** single channel
- **Color:** three channel (RGB)
- Spatial and hierarchical correlation structures



UPPSALA
UNIVERSITET

- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

MNIST example

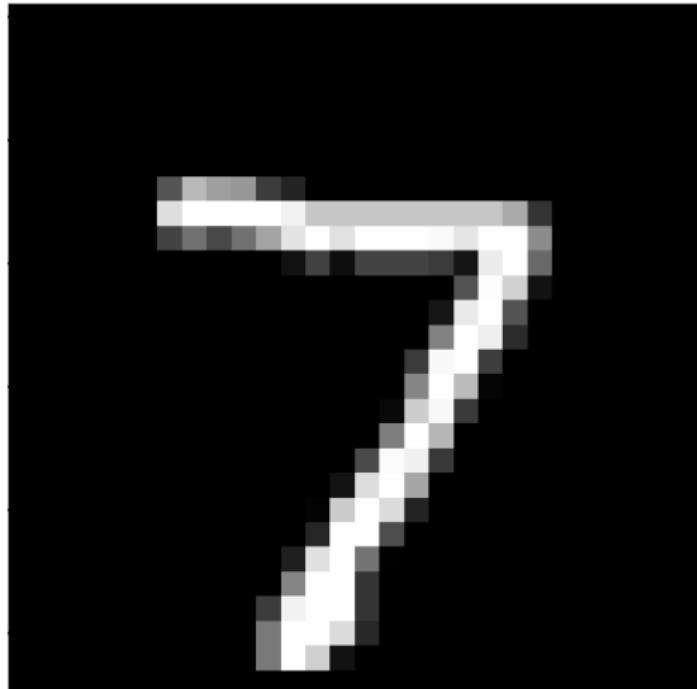


Figure: Example from the MNIST dataset (28 by 28 pixels)



- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

How to train models for images?

- We want to learn **representations** of parts of images

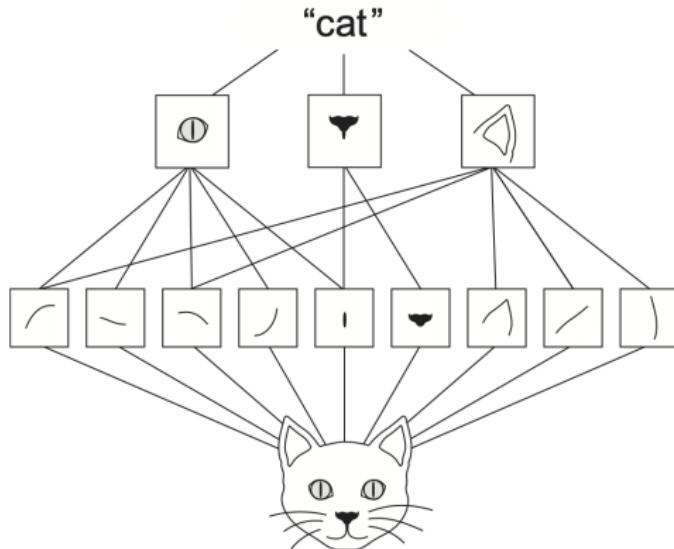


Figure: The representations of a cat (Chollet and Allair, 2018, Fig 5.2)



- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

How to train models for images?

- We want to learn **representations** of parts of images

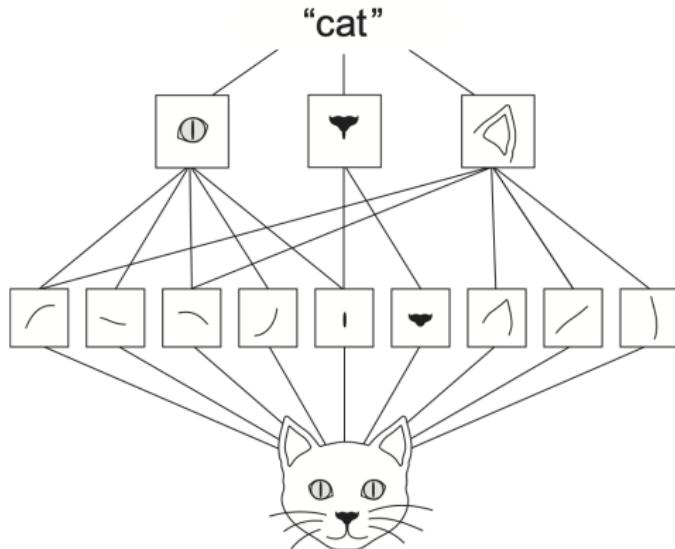


Figure: The representations of a cat (Chollet and Allair, 2018, Fig 5.2)

- CNN uses **Convolutional Layers** to learn **parameter efficient** representations



- Previous assignments
- The Mini-Project
- **Introduction**
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Learning Representations for Images (again)

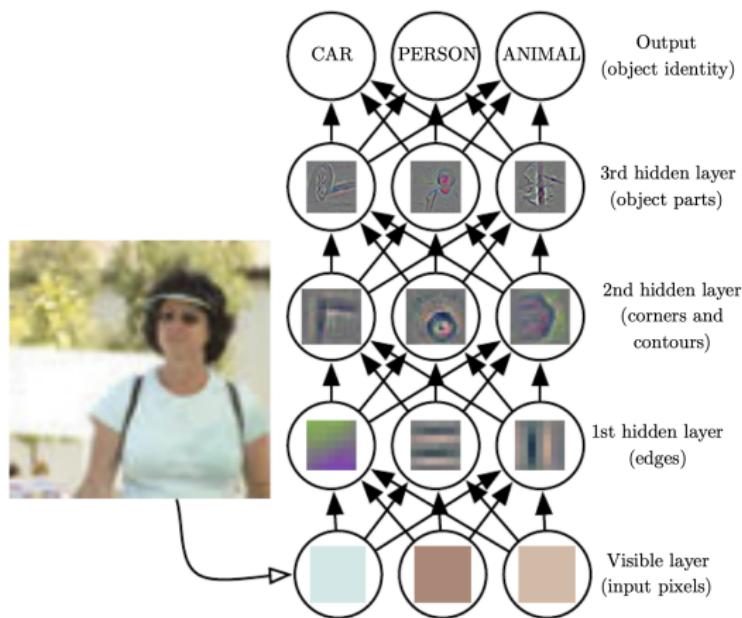


Figure: Learning representations for images (Goodfellow et al, 2017, Fig. 1.2)



UPPSALA
UNIVERSITET

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Section 4

Convolution



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolution

- Different definitions are common, one example:

$$y(t) = \int x(\tau)k(t - \tau)d\tau = (x * k)(t)$$

- Intuition: "Weighting together two functions"
- In a convolutional layer:
 1. $x(t)$: Input
 2. $k(t)$: Kernel, filter, "feature"
 3. $y(t)$: Output, feature map

DEMO



UPPSALA UNIVERSITET

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Discrete Convolution

- If t is discrete (as in a grid):

$$y(t) = (x * k)(t) = \sum_{\tau=-\infty}^{\infty} x(\tau)k(t-\tau)$$



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Discrete Convolution

- If t is discrete (as in a grid):

$$y(t) = (x * k)(t) = \sum_{\tau=-\infty}^{\infty} x(\tau)k(t-\tau)$$

- In the case of images we have 2 discrete dimensions

$$Y(i, j) = (X * K) \sum_m \sum_n X(m, n)K(i - m, j - n)$$

- Sometimes the cross-correlation is called convolution:

$$Y(i, j) = (X * K) \sum_m \sum_n X(m, n)K(i + m, j + n)$$

1. $X(i, j)$: Input (2D)
2. $K(i, j)$: Kernel, filter, "feature" (2D)
3. $Y(i, j)$: Output, feature map (2D)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolution of Images: 2D

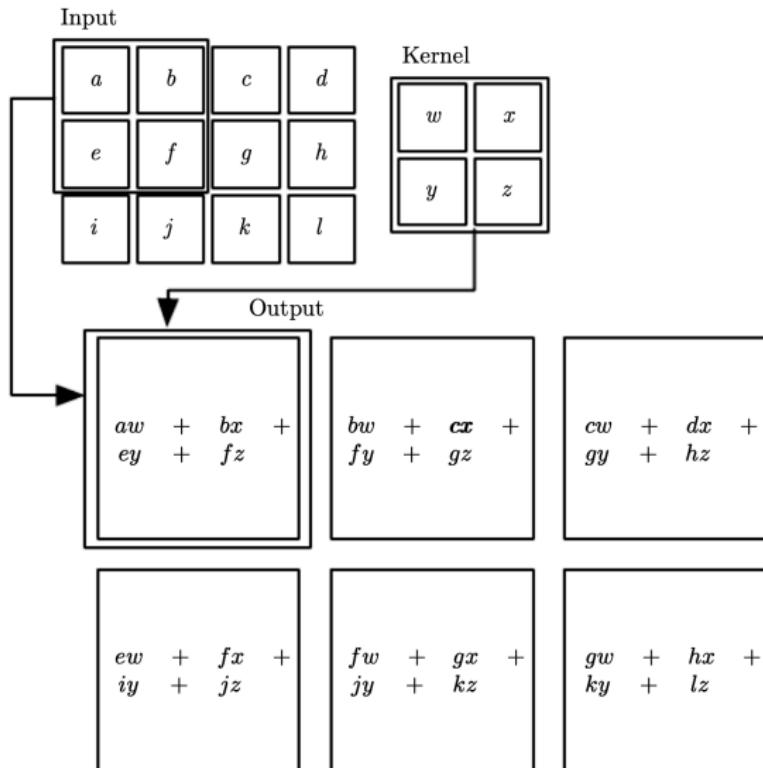


Figure: Convolution for an Image (Goodfellow et al, 2017, Fig. 9.1)



UPPSALA
UNIVERSITET

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolution of images: Example

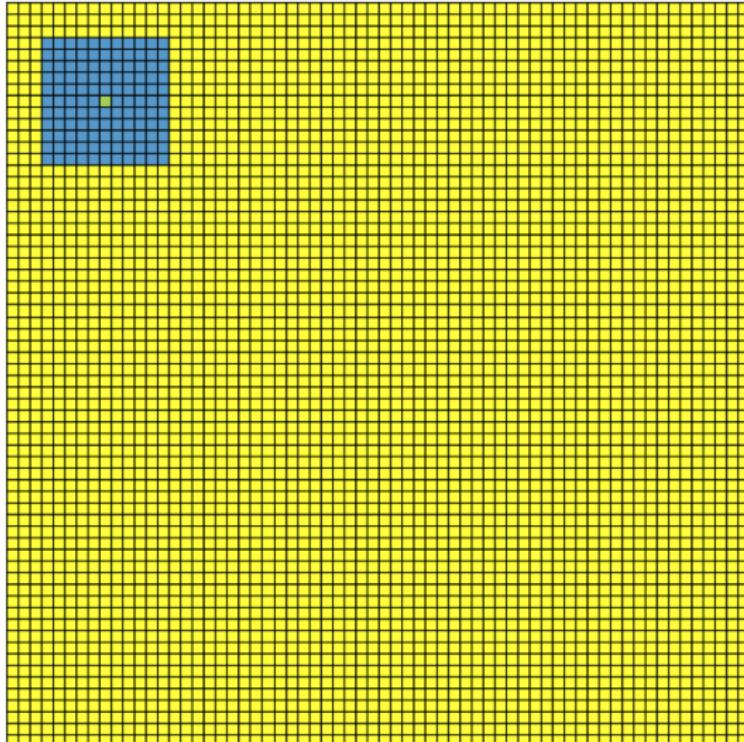


Figure: Convolution example.



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolution of images: Examples

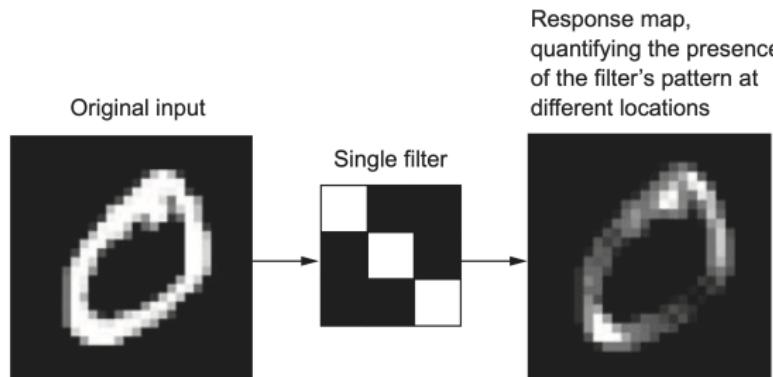


Figure: Convolution for an Image (Chollet and Allaire, 2018, Fig. 5.3)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

$$X = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}, K = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$$



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolution of images: Example

$$X = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}, K = \begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$Y = \begin{bmatrix} 0 & 0 & 2 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 1 & 2 & 0 \end{bmatrix},$$



UPPSALA
UNIVERSITET

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- **Convolutional Neural Networks**
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Section 5

Convolutional Neural Networks



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- **Convolutional Neural Networks**
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolutional Neural Networks

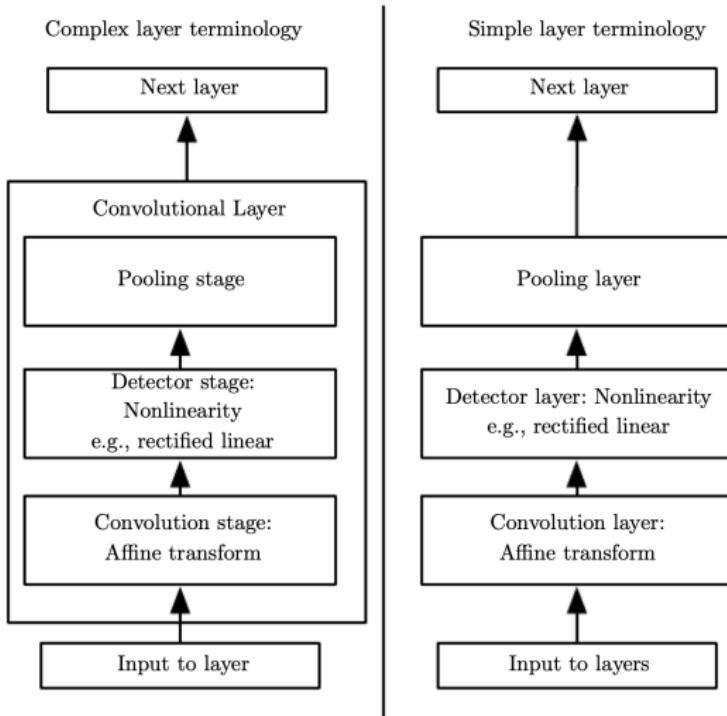


Figure: Convolution layer (Goodfellow et al, 2018, Fig. 9.7)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- **Convolutional Neural Networks**
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolutional Neural Networks

- Most convolutional neural networks have:
 1. Many convolutional layers
 2. More kernels higher up in the network
 3. A classification head (usually a feed-forward neural network)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- **Convolutional Neural Networks**
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolutional Neural Networks

- Most convolutional neural networks have:
 1. Many convolutional layers
 2. More kernels higher up in the network
 3. A classification head (usually a feed-forward neural network)
- Benefits:
 1. Few(er) parameters (filters)
 2. Captures **local structures**
 3. Efficient computations



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolutional Neural Networks

- Most convolutional neural networks have:
 1. Many convolutional layers
 2. More kernels higher up in the network
 3. A classification head (usually a feed-forward neural network)
- Benefits:
 1. Few(er) parameters (filters)
 2. Captures **local structures**
 3. Efficient computations
- How to choose filters?
 1. Before: **manually handcrafted**



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolutional Neural Networks

- Most convolutional neural networks have:
 1. Many convolutional layers
 2. More kernels higher up in the network
 3. A classification head (usually a feed-forward neural network)
- Benefits:
 1. Few(er) parameters (filters)
 2. Captures **local structures**
 3. Efficient computations
- How to choose filters?
 1. Before: **manually handcrafted**
 2. Now: **learn the filters**



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolution layer

- **Input:** Data or Feature Maps



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolution layer

- **Input:** Data or Feature Maps
- **Parameters:**
 - N filters/kernels of size $m \times m$
 - N bias terms (one per filter)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolution layer

- **Input:** Data or Feature Maps
- **Parameters:**
 - N filters/kernels of size $m \times m$
 - N bias terms (one per filter)
- **Activation functions:** Applied element wise on feature maps



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolution layer

- **Input:** Data or Feature Maps
- **Parameters:**
 - N filters/kernels of size $m \times m$
 - N bias terms (one per filter)
- **Activation functions:** Applied element wise on feature maps
- **Output:** Feature Maps



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolution layer

- **Input:** Data or Feature Maps
- **Parameters:**
 - N filters/kernels of size $m \times m$
 - N bias terms (one per filter)
- **Activation functions:** Applied element wise on feature maps
- **Output:** Feature Maps
- In Keras:
`layer_conv_2d(filters = 32, kernel_size = c(3,3), activation = "relu", input_shape = c(32,32,3))`



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Padding

- Handling edges
- *Padding*: add 0 around the image
- Necessary to **keep size** of feature maps



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Padding

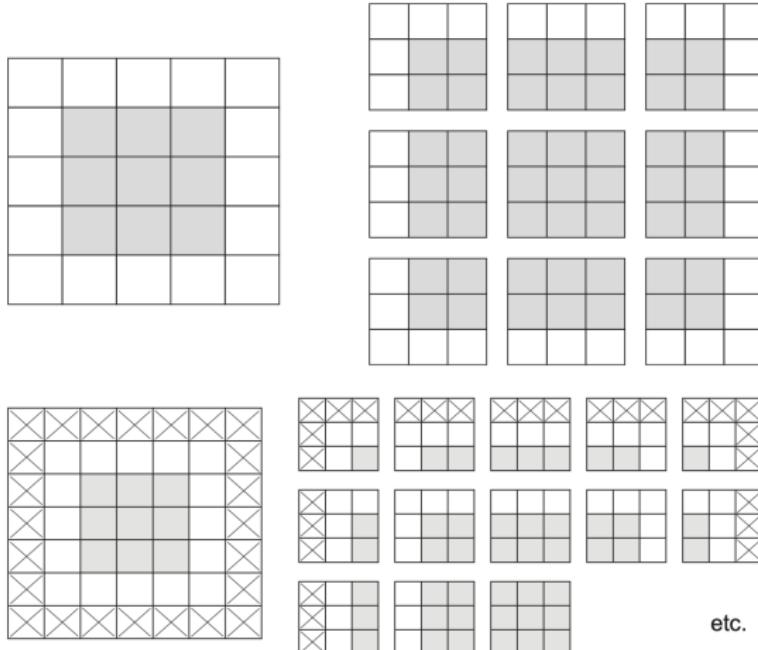


Figure: Padding and valid edge handling (Chollet and Allair (2018), Fig. 5.5, 5.6)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Stride

- Skip every n th pixel
- Reduces the computations

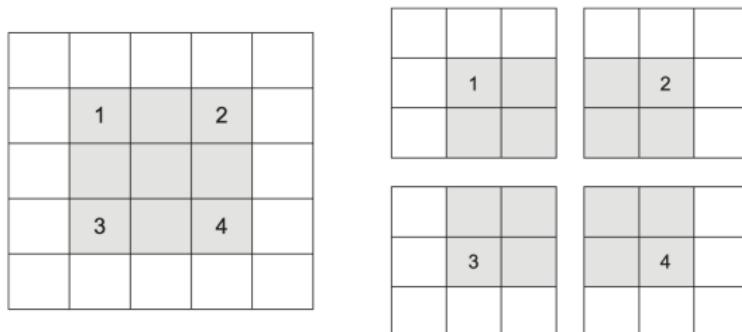


Figure: Strides (Chollet and Allair (2018), Fig. 5.5, 5.6)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Why Convolution Layers?

- Captures local spatial structure
- Reduces the number of parameters (parameter sharing)
 1. The number and size of filters
 2. We use the same filters everywhere



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Why Convolution Layers?

- Captures local spatial structure
- Reduces the number of parameters (parameter sharing)
 1. The number and size of filters
 2. We use the same filters everywhere
- Example: a 1 megapixel image (1000×1000 pixels)
 1. Dense network with 100 nodes: **100M** parameters
 2. CNN network with 100 3×3 filters: **1000** parameters
(900 from filters, 100 bias terms)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Convolution Neural Nets

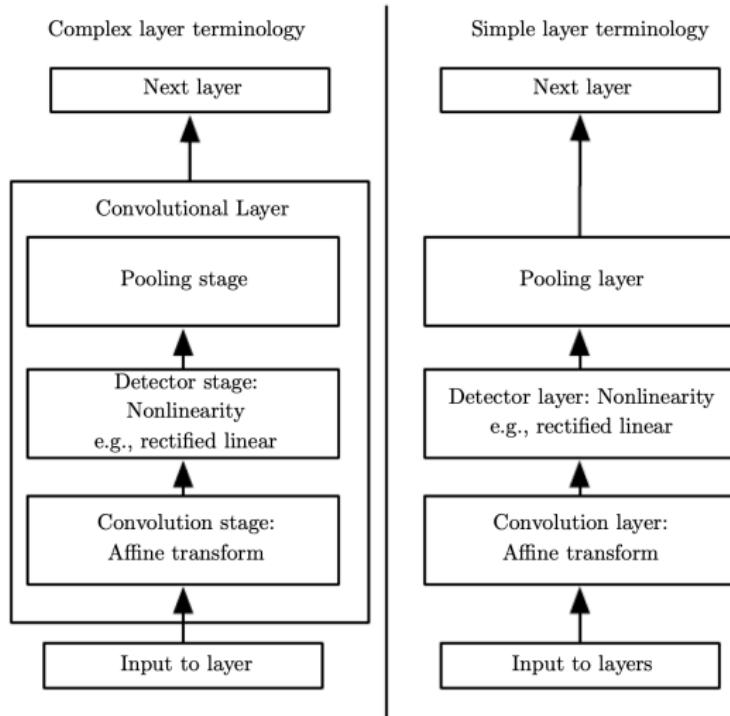


Figure: Convolution layer (Goodfellow et al, 2018, Fig. 9.7)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Detector stage

- Remember, in feed-forward networks: $\mathbf{h} = \sigma(\mathbf{XW} + b)$
- In CNN:
 1. \mathbf{W} is the filter



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Detector stage

- Remember, in feed-forward networks: $\mathbf{h} = \sigma(\mathbf{XW} + b)$
- In CNN:
 1. \mathbf{W} is the filter
 2. \mathbf{X} is the input feature map



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Detector stage

- Remember, in feed-forward networks: $\mathbf{h} = \sigma(\mathbf{XW} + b)$
- In CNN:
 1. \mathbf{W} is the filter
 2. \mathbf{X} is the input feature map
 3. \mathbf{XW} is the convolutional feature map



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Detector stage

- Remember, in feed-forward networks: $\mathbf{h} = \sigma(\mathbf{XW} + b)$
- In CNN:
 1. \mathbf{W} is the filter
 2. \mathbf{X} is the input feature map
 3. \mathbf{XW} is the convolutional feature map
 4. b is a bias (one per filter)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Detector stage

- Remember, in feed-forward networks: $\mathbf{h} = \sigma(\mathbf{XW} + b)$
- In CNN:
 1. \mathbf{W} is the filter
 2. \mathbf{X} is the input feature map
 3. \mathbf{XW} is the convolutional feature map
 4. b is a bias (one per filter)
 5. σ is the activation function (usually a ReLU)



- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - **The Pooling Layer**
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- We take a function f that return one value per pooling kernel



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - [The Pooling Layer](#)
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Pooling layer

- We take a function f that return one value per pooling kernel
- Most commonly $f = \max$
- Commonly a 2×2 pooling kernel with stride 2
- **Why?** Reduce the size of feature map, but keep the activation



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Pooling layer

- We take a function f that return one value per pooling kernel
- Most commonly $f = \max$
- Commonly a 2×2 pooling kernel with stride 2
- **Why?** Reduce the size of feature map, but keep the activation
- In Keras:
`layer_max_pooling_2d(pool_size = c(2, 2))(`



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Max Pooling

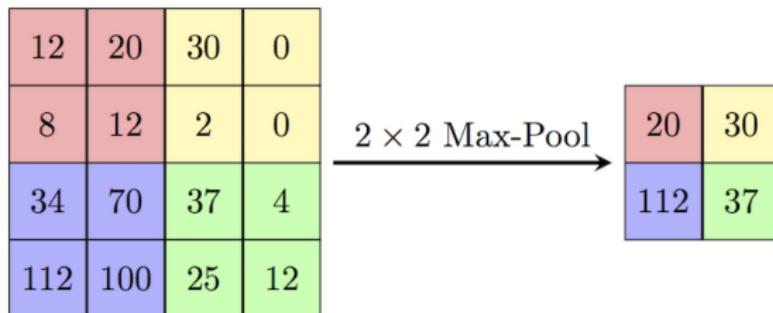


Figure: Strides (Computer Science Wikipedia)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Using pooling to learn invariances

- pooling over spatial positions: invariant to translation
- pooling over different filters: invariant to transformations

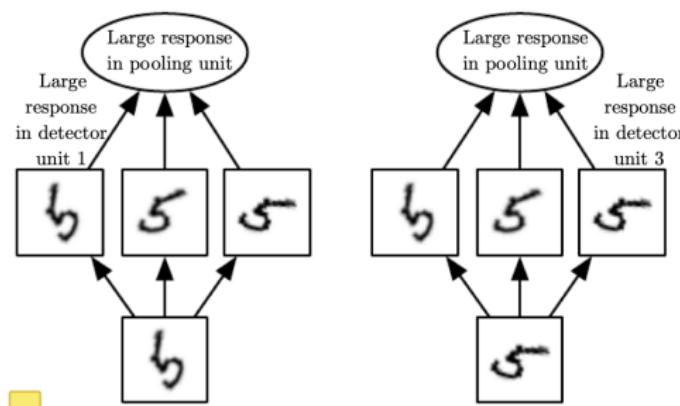


Figure: Learning invariances (Goodfellow et al., 2017, Fig. 9.9)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Data Augmentation



Figure: Data Augmentation (Chollet and Allair, 2018, Fig 5.10)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Data Augmentation



Figure: Data Augmentation (Chollet and Allair, 2018, Fig 5.10)

- Can be done directly in Keras (data generator)



- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
-
- AlexNet (2012), 5 convolutional layers
 - VGG16 (2014), 16 convolutional layers
 - ResNet (2015), 152 convolutional layers



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

VGG16

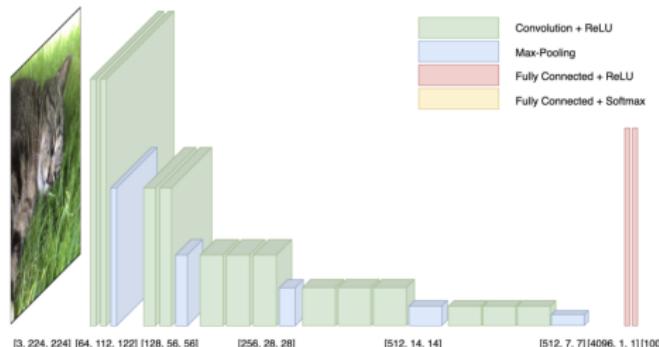


Figure 2.4: Illustration of the data flow through the network VGG16. Data size, of the format $[c, h, w]$, is shown for the input image, output of each max-pooling layer, output after the first two fully connected layers, and the final network output.

Jesper Westell, Multi-Task Learning using Road Surface Condition Classification and Road Scene Semantic Segmentation, LIU-IMT-TFK-A-19/570-SE



UPPSALA
UNIVERSITET

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Section 6

Transfer learning



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Transfer learning

- "Transfer knowledge between problems"
- Learning representations in P_1 will aid generalization in P_2



- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
-
- "Transfer knowledge between problems"
 - Learning representations in P_1 will aid generalization in P_2
 - A Bayesian perspective: A strong prior



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

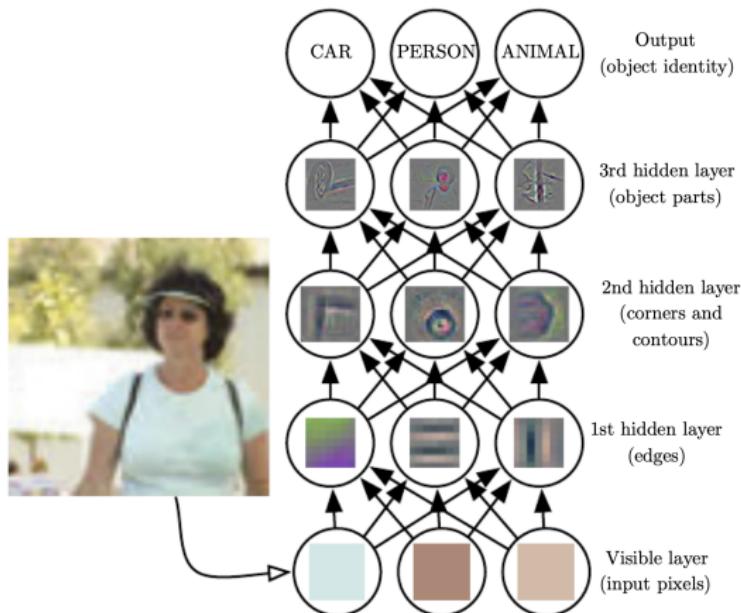


Figure: Learning representations can be crucial (Goodfellow et al, 2017, Fig. 1.2)



- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- In practice: Transfer/reuse learned weights or rather some weights



- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- In practice: Transfer/reuse learned weights or rather some weights
 - Use (large) pre-trained models for smaller problems
 - A reason for the success of CNN



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Transfer learning

- In practice: Transfer/reuse **learned weights** or rather some weights
- Use (large) **pre-trained** models for smaller problems
- A reason for the success of CNN
- Two types of transfer learning in Neural Networks:
 - Feature extraction (use pre-trained networks for features)
 - Fine Tuning (adapt pre-trained features)



UPPSALA UNIVERSITET

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Feature Extraction

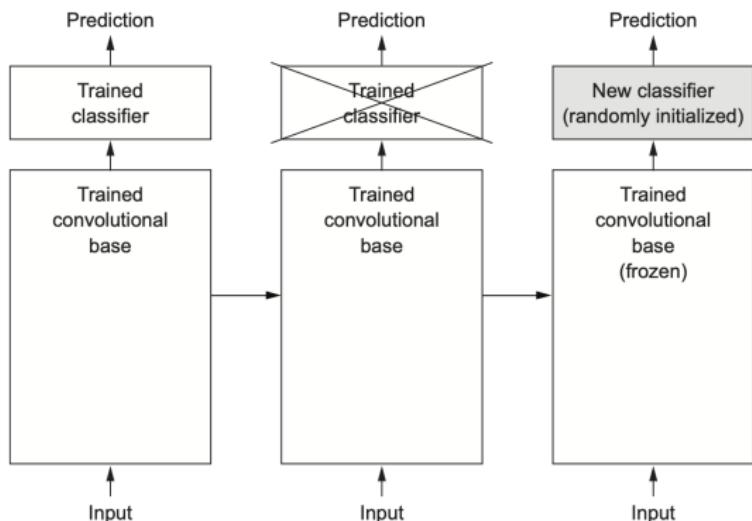


Figure: Using convnets as base for feature extraction (Chollet and Allair, 2018, Fig 5.12)



UPPSALA UNIVERSITET

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Fine-Tuning

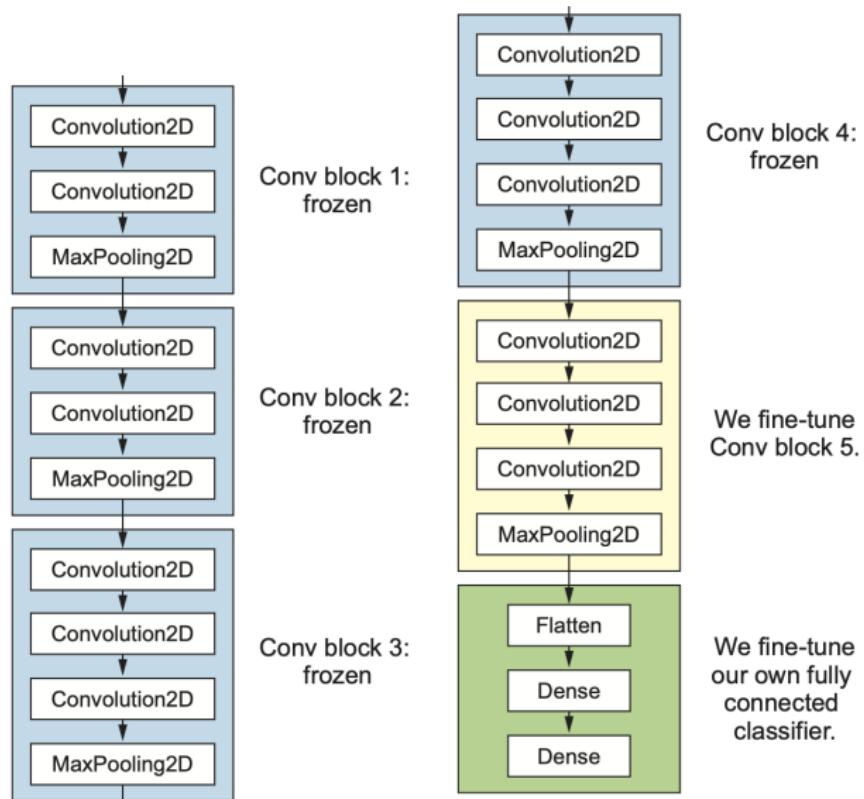


Figure: Finetuning a convolutional base (Chollet and Allair, 2018,
Fig 5.15)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Transfer learning

- Catastrophic forgetting



- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- Catastrophic forgetting
 - Domain adaptation: Same problem but at different input dataset
 - (e.g. language models for legal/medical/political data)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Transfer learning

- Catastrophic forgetting
- Domain adaptation: Same problem but at different input dataset
(e.g. language models for legal/medical/political data)
- Concept drift: Similar problem



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Transfer learning

- Catastrophic forgetting
- Domain adaptation: Same problem but at different input dataset
(e.g. language models for legal/medical/political data)
- Concept drift: Similar problem
- Previously, popular with unsupervised pre-training.



UPPSALA
UNIVERSITET

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Section 7

Practical Methodology



UPPSALA
UNIVERSITET

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Practical Methodology

1. Determine your goals



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Practical Methodology

1. Determine your goals
2. Setup your baseline
(establish a working end-to-end pipeline as soon as possible)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Practical Methodology

1. Determine your goals
2. Setup your baseline
(establish a working end-to-end pipeline as soon as possible)
3. Diagnose your networks performance



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Practical Methodology

1. Determine your goals
 2. Setup your baseline
(establish a working end-to-end pipeline as soon as possible)
 3. Diagnose your networks performance
 4. Make incremental improvements
-
- **General idea:** Increase data and model capacity until goal is reached





- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Practical Methodology

1. Determine your goals
 2. Setup your baseline
(establish a working end-to-end pipeline as soon as possible)
 3. Diagnose your networks performance
 4. Make incremental improvements
-
- General idea: Increase data and model capacity until goal is reached
 - End goal: Good enough performance on test set





1. Determine your goals

- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- Why are you building a model?





- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
-
- Why are you building a model?
 - Set up the metric based on the **overall goal** of the system!
This may need multiple metrics.



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

1. Determine your goals

- Why are you building a model?
- Set up the metric based on the **overall goal** of the system!
This may need multiple metrics.
- What is good enough? Remember the **Bayes error**!



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

1. Determine your goals

- Why are you building a model?
- Set up the metric based on the **overall goal** of the system!
This may need multiple metrics.
- What is good enough? Remember the **Bayes error**!
- What performance can you expect?



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

1. Determine your goals

- Why are you building a model?
- Set up the metric based on the **overall goal** of the system!
This may need multiple metrics.
- What is good enough? Remember the **Bayes error**!
- What performance can you expect?
- Some errors are worse than others, e.g. spam filters.



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

1. Determine your goals

- Why are you building a model?
- Set up the metric based on the **overall goal** of the system!
This may need multiple metrics.
- What is good enough? Remember the **Bayes error**!
- What performance can you expect?
- Some errors are worse than others, e.g. spam filters.
- Handling of uncertain predictions:
- **Coverage:** How large proportions can the system predict?



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

1. Determine your goals

- Why are you building a model?
- Set up the metric based on the **overall goal** of the system!
This may need multiple metrics.
- What is good enough? Remember the **Bayes error**!
- What performance can you expect?
- Some errors are worse than others, e.g. spam filters.
- Handling of uncertain predictions:
- **Coverage:** How large proportions can the system predict?
- Manual curation can be faster and easier.



- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- Start with...
 - the most simple possible model (logistic regression)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

2. Setup your baseline

- Start with...
 - the most simple possible model (logistic regression)
 - previous approaches/baselines



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

2. Setup your baseline

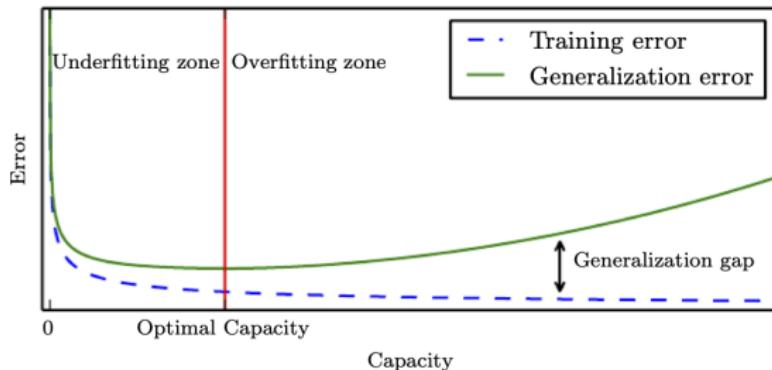
- Start with...
 - the most simple possible model (logistic regression)
 - previous approaches/baselines
 - a simple neural network that is common in the domain/defaults
(CNN for images, Adam as optimizer)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Remember: Statistical Learning Theory

Figure: Test, training, and model complexity (Goodfellow et al, 2017, Figure 5.3)





- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- High training loss: Training data not fully used
Neural network generally performs best when training error is low



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

3. Diagnose and improve

- High training loss: Training data not fully used
Neural network generally performs best when training error is low
- High test loss: Low data quality, i.e. large Bayes error?



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

3. Diagnose and improve

- High training loss: Training data not fully used
Neural network generally performs best when training error is low
- High test loss: Low data quality, i.e. large Bayes error?
- Low training error and high test error: Common situation



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

3. Diagnose and improve

- High training loss: Training data not fully used
Neural network generally performs best when training error is low
- High test loss: Low data quality, i.e. large Bayes error?
- Low training error and high test error: Common situation
 - You can always improve by gather more data, or
 - Regularize to optimal capacity



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

3. Diagnose and improve

- High training loss: Training data not fully used
Neural network generally performs best when training error is low
- High test loss: Low data quality, i.e. large Bayes error?
- Low training error and high test error: Common situation
 - You can always improve by gather more data, or
 - Regularize to optimal capacity
- How to know the improvements of additional data?



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Learning curve

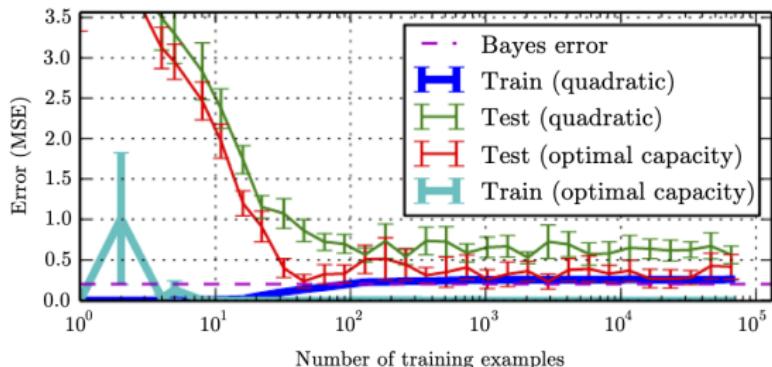


Figure: Learning curve to assess the need for more data (Goodfellow et al., 2017, Fig 5.4)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

3. Diagnosing and improving your model: Regularization

- Best performance: Larger model that is regularized well.
- Warning: Avoid **the algorithm rabbit hole**



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

3. Diagnosing and improving your model: Regularization

- Best performance: Larger model that is regularized well.
- Warning: Avoid **the algorithm rabbit hole**
- Adapt hyperparameters to get **optimal capacity**



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

3. Diagnosing and improving your model: Regularization

- Best performance: Larger model that is regularized well.
- Warning: Avoid **the algorithm rabbit hole**
- Adapt hyperparameters to get **optimal capacity**
- Hyperparameter Search Goal:
adjust the model capacity to match the complexity of the task.





- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

3. Diagnosing and improving your model: Regularization

- Best performance: Larger model that is regularized well.
- Warning: Avoid **the algorithm rabbit hole**
- Adapt hyperparameters to get **optimal capacity**
- Hyperparameter Search Goal:
adjust the model capacity to match the complexity of the task.
- Marginal hyperparameter has a U-shaped error function (ideally)



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

3. Diagnosing and improving your model: Regularization

- Best performance: Larger model that is regularized well.
- Warning: Avoid **the algorithm rabbit hole**
- Adapt hyperparameters to get **optimal capacity**
- Hyperparameter Search Goal:
adjust the model capacity to match the complexity of the task.
- Marginal hyperparameter has a U-shaped error function (ideally)
- Neural Networks Steps:
 1. Get good training error. E.g. by tune learning rate and increasing capacity



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

3. Diagnosing and improving your model: Regularization

- Best performance: Larger model that is regularized well.
- Warning: Avoid **the algorithm rabbit hole**
- Adapt hyperparameters to get **optimal capacity**
- Hyperparameter Search Goal:
adjust the model capacity to match the complexity of the task.
- Marginal hyperparameter has a U-shaped error function (ideally)
- Neural Networks Steps:
 1. Get good training error. E.g. by tune learning rate and increasing capacity
 2. Tune hyperparameters (regularization):
Requires monitoring both training and test error



UPPSALA
UNIVERSITET

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Hyper-parameter optimization

- Neural networks has many hyperparameters
- What is a hyperparameter in a neural network?



UPPSALA UNIVERSITET

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Hyper-parameter optimization

- Neural networks has many hyperparameters
- What is a hyperparameter in a neural network?
- Grid search:
 - Setup a grid of potential values
 - For 1-4 hyperparameter: grid search can work well
 - Usually iterative/repeated grid search is best: start with three values, and work iteratively
 - Grows exponentially with the number of parameters



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Hyper-parameter optimization

- Neural networks has many hyperparameters
- What is a hyperparameter in a neural network?
- Grid search:
 - Setup a grid of potential values
 - For 1-4 hyperparameter: grid search can work well
 - Usually iterative/repeated grid search is best: start with three values, and work iteratively
 - Grows exponentially with the number of parameters
- Random search
 - Specify a marginal distribution for each hyperparameter (additional work)
 - Common: uniform on log scale
 - Can also be done iteratively
 - Can be exponentially more efficient
 - Generally reduce the error faster in setting with many hyper parameters



Diagnosis and Debugging

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology
- Visualize the model predictions





Diagnosis and Debugging

- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology
 - Visualize the model predictions
 - Analyze the worst errors: **why?**



- Previous assignments
 - The Mini-Project
 - Introduction
 - Convolution
 - Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
 - Transfer learning
 - Practical Methodology
- Visualize the model predictions
 - Analyze the worst errors: **why?**
 - Use train and test error as a diagnostic: **Can you overfit the data?**



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Diagnosis and Debugging

- Visualize the model predictions
- Analyze the worst errors: **why?**
- Use train and test error as a diagnostic: **Can you overfit the data?**
- Use test suites: Can you get known results on a toy data.
Both training error and derivatives



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Diagnosis and Debugging

- Visualize the model predictions
- Analyze the worst errors: **why?**
- Use train and test error as a diagnostic: **Can you overfit the data?**
- Use test suites: Can you get known results on a toy data.
Both training error and derivatives
- Monitor the gradients



- Previous assignments
- The Mini-Project
- Introduction
- Convolution
- Convolutional Neural Networks
 - The Convolution Layer
 - The Pooling Layer
 - Regularization
 - Examples
- Transfer learning
- Practical Methodology

Diagnosis and Debugging

- Visualize the model predictions
- Analyze the worst errors: **why?**
- Use train and test error as a diagnostic: **Can you overfit the data?**
- Use test suites: Can you get known results on a toy data.
Both training error and derivatives
- Monitor the gradients
- Monitor activation function statistics. Are some never activated?