In this lecture series, leading research scientists from leading AI research lab, DeepMind, will give 12 lectures on an exciting selection of topics in Deep Learning, ranging from the fundamentals of training neural networks via advanced ideas around memory, attention, and generative modelling to the important topic of responsible innovation.

Please join us for a deep dive lecture series into Deep Learning!

#UCLxDeepMind
General information

Exits:
At the back, the way you came in

Wifi:
UCL guest
Sander Dieleman is a Research Scientist at DeepMind in London, UK, where he has worked on the development of AlphaGo and WaveNet. He was previously a PhD student at Ghent University, where he conducted research on feature learning and deep learning techniques for learning hierarchical representations of musical audio signals. During his PhD he also developed the deep learning library Lasagne and won solo and team gold medals respectively in Kaggle's "Galaxy Zoo" competition and the first National Data Science Bowl. In the summer of 2014, he interned at Spotify in New York, where he worked on implementing audio-based music recommendation using deep learning on an industrial scale.
In the past decade, convolutional neural networks have revolutionised computer vision. In this lecture, we will take a closer look at convolutional network architectures through several case studies, ranging from the early 90's to the current state of the art. We will review some of the building blocks that are in common use today, discuss the challenges of training deep models, and strategies for finding effective architectures, with a focus on image recognition.
Convolutional Neural Networks for Image Recognition

Sander Dieleman
Plan for this lecture

01 Background

02 Building blocks

03 Convolutional neural networks

04 Going deeper: case studies

05 Advanced topics

06 Beyond image recognition
1 Background
Last week: neural networks
Loss
Linear
Node
Cross entropy
Target
Softmax
Data
Linear
Sigmoid
How can we feed images to a neural network?
Linear > Sigmoid > Linear > Softmax > Cross entropy

Target

Data

Linear

Cross entropy
A digital image is a 2D grid of pixels.
Neural networks for images

A digital image is a 2D grid of pixels.
Neural networks for images

A digital image is a 2D grid of pixels.

A neural network expects a vector of numbers as input.
Neural networks for images

A digital image is a 2D grid of pixels.

A neural network expects a vector of numbers as input.
Neural networks for images

A digital image is a 2D grid of pixels.

A neural network expects a vector of numbers as input.
Locality and translation invariance

**Locality**: nearby pixels are more strongly correlated

**Translation invariance**: meaningful patterns can occur anywhere in the image
Content

Images with repeating patterns

Goal

Demonstrate the locality and translation invariance properties of images in an exaggerated way
Data modalities with topological structure: audio, text, graphs.

Goal: Highlight that many other types of data have similar structure that we can take advantage of in neural network architectures.
Taking advantage of topological structure
Taking advantage of topological structure

Weight sharing: use the same network parameters to detect local patterns at many locations in the image
Taking advantage of topological structure

**Weight sharing**: use the same network parameters to detect local patterns at many locations in the image

**Hierarchy**: local low-level features are composed into larger, more abstract features

edges and textures → object parts → objects
Data drives research

Availability of large datasets has led to major breakthroughs within deep learning and beyond. Goal: Set the stage for the history of convnets and how the ImageNet challenge drove this research for many years.
The ImageNet challenge

- Major computer vision benchmark
- Ran from 2010 to 2017
- 1.4M images, 1000 classes
- Image classification

Want to learn more?

Top-5 classification error rate of the competition winners

- 2010: 28%
- 2011: 26%
- 2012: 16%
- 2013: 12%
- 2014: 7%
- 2015: 4%
- 2016: 3%
- 2017: 2%
Traditional computer vision techniques
VGGNet and GoogLeNet
ResNet
From fully connected to locally connected
From fully connected to locally connected
From fully connected to locally connected

\[ y = \sum_{i \in \text{image}} w_i x_i + b \]
From fully connected to locally connected

\[ y = \sum_{i \in 3 \times 3} w_i x_i + b \]

locally-connected units
3 × 3 receptive field
From locally connected to convolutional

$$y = w \ast x + b$$

convolutional units
3x3 receptive field
From locally connected to convolutional

Receptive field

Feature map
Implementation: the convolution operation

The kernel slides across the image and produces an output value at each position.
Implementation: the convolution operation

The kernel slides across the image and produces an output value at each position.
Implementation: the convolution operation

The kernel slides across the image and produces an output value at each position.
Implementation: the convolution operation

The kernel slides across the image and produces an output value at each position.
Implementation: the convolution operation

The kernel slides across the image and produces an output value at each position.
Implementation: the convolution operation

The kernel slides across the image and produces an output value at each position.
Implementation: the convolution operation

The kernel slides across the image and produces an output value at each position.

The kernel slides across the image and produces an output value at each position.
Implementation: the convolution operation

We convolve multiple kernels and obtain multiple feature maps or channels.
Inputs and outputs are tensors
Inputs and outputs are tensors
Inputs and outputs are tensors
Variants of the convolution operation

**Valid convolution**: output size = input size – kernel size + 1
Variants of the convolution operation

**Full convolution**: output size = input size + kernel size – 1
Variants of the convolution operation

**Same convolution**: output size = input size
Variants of the convolution operation

**Strided convolution:** kernel slides along the image with a step $> 1$
Variants of the convolution operation

Strided convolution: kernel slides along the image with a step > 1
Variants of the convolution operation

**Strided convolution:** kernel slides along the image with a step > 1
Variants of the convolution operation

**Strided convolution**: kernel slides along the image with a step > 1
Variants of the convolution operation

Dilated convolution: kernel is spread out, step > 1 between kernel elements
Variants of the convolution operation

Dilated convolution: kernel is spread out, step > 1 between kernel elements
Variants of the convolution operation

Dilated convolution: kernel is spread out, step > 1 between kernel elements
Variants of the convolution operation

Depthwise convolution: each output channel is connected only to one input channel
Pooling: compute mean or max over small windows to reduce resolution
Pooling: compute mean or max over small windows to reduce resolution
3 Convolutional neural networks
Stacking the building blocks

- CNNs or “convnets”
- Up to 100s of layers
- Alternate convolutions and pooling to create a hierarchy
Recap: neural networks as computational graphs

- **input**
- **computation**
- **loss**
- **parameters**
Simplified diagram: implicit parameters and loss

input

computation
Simplified diagram: implicit parameters and loss

- Input
- Computation
Computational building blocks of convnets

- Input
- Nonlinearity
- Convolution
- Pooling
- Fully connected
4 Going deeper: Case studies
LeNet-5 (1998)

Architecture of LeNet-5, a convnet for handwritten digit recognition

Want to learn more?
Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.
Gradient-based learning applied to document recognition
AlexNet (2012)

Architecture: 8 layers, ReLU, dropout, weight decay

Infrastructure: large dataset, trained 6 days on 2 GPUs

Want to learn more?
Krizhevsky, A.; Sutskever, I.; Hinton, G.E. 
*ImageNet classification with deep convolutional neural networks*
AlexNet (2012)

Figure from Krizhevsky et al. (2012)
AlexNet (2012)

Input image:
→ 224×224×3
Layer 1 convolution:
kernel $11 \times 11$, 96 channels, stride 4
$\rightarrow 56 \times 56 \times 96$
AlexNet (2012)

ReLU
AlexNet (2012)

Max-pooling:
window 2×2
→ 28×28×96
Layer 8 fully connected: $\rightarrow 1000$
AlexNet (2012)
AlexNet (2012)

- Input: 224 x 224 x 3
- Convolutional layer: 56 x 56 x 96, ReLU
- Convolutional layer: 28 x 28 x 256, ReLU
- Convolutional layer: 14 x 14 x 256, ReLU
- Convolutional layer: 14 x 14 x 384, ReLU
- Convolutional layer: 14 x 14 x 384, ReLU
- Fully connected layer: 4096
- Fully connected layer: 4096
- Convolutional layer: 7 x 7 x 256, ReLU
- Convolutional layer: 14 x 14 x 256, ReLU
- Softmax output: 1000
Deeper is better

- Each layer is a linear classifier by itself
- More layers – more nonlinearities
- What limits the number of layers in convnets?
VGGNet (2014): building very deep convnets

Want to learn more?

Simonyan, K.; Zisserman, A.
Very deep convolutional networks for large-scale image recognition

Stack many convolutional layers before pooling
Use “same” convolutions to avoid resolution reduction
VGGNet (2014): stacking $3 \times 3$ kernels

Architecture: up to 19 layers, $3 \times 3$ kernels only, “same” convolutions

Infrastructure: trained for 2–3 weeks on 4 GPUs (data parallelism)
VGGNet (2014): error plateaus after 16 layers
Challenges of depth

- Computational complexity
- Optimisation difficulties
Improving optimisation

- Careful initialisation
- Sophisticated optimisers
- Normalisation layers
- Network design
GoogLeNet (2014)

Figure from Szegedy et al. (2015)

Want to learn more?

Szegedy, C. et al. 
*Going deeper with convolutions* IEEE conference on computer vision and pattern recognition (2015)
Batch normalisation

**Input:** Values of $x$ over a mini-batch: $B = \{x_1...x_m\}$; 
Parameters to be learned: $\gamma, \beta$

**Output:** $\{y_i = \text{BN}_{\gamma,\beta}(x_i)\}$

$$
\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i \quad \text{ // mini-batch mean}
$$

$$
\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^{m} (x_i - \mu_B)^2 \quad \text{ // mini-batch variance}
$$

$$
\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad \text{ // normalize}
$$

$$
y_i \leftarrow \gamma \hat{x}_i + \beta \equiv \text{BN}_{\gamma,\beta}(x_i) \quad \text{ // scale and shift}
$$

---

**Want to learn more?**


---

**Reduces sensitivity to initialisation**

Introduces stochasticity and acts as a **regulariser**
Batch normalisation

Figure from Ioffe et al. (2015)

Residual connections facilitate training deeper networks

Want to learn more?

ResNet (2015): different flavours

ResNet V2 (bottom) avoids all nonlinearities in the residual pathway

Want to learn more?
He, K. et al. *Identity mappings in deep residual networks*. European conference on computer vision (2016)
## ResNet (2015): up to 152 layers

<table>
<thead>
<tr>
<th>layer name</th>
<th>output size</th>
<th>18-layer</th>
<th>34-layer</th>
<th>50-layer</th>
<th>101-layer</th>
<th>152-layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv1</td>
<td>112×112</td>
<td>7×7, 64, stride 2</td>
<td>3×3 max pool, stride 2</td>
<td>3×3 max pool, stride 2</td>
<td>3×3 max pool, stride 2</td>
<td>3×3 max pool, stride 2</td>
</tr>
<tr>
<td>conv2_x</td>
<td>56×56</td>
<td>×2 [3×3, 64] [3×3, 64]</td>
<td>×3 [3×3, 64] [3×3, 64]</td>
<td>×3 [1×1, 64] [3×3, 64]</td>
<td>×3 [1×1, 64] [3×3, 64]</td>
<td>×3 [1×1, 64] [3×3, 64]</td>
</tr>
<tr>
<td>conv3_x</td>
<td>28×28</td>
<td>×2 [3×3, 128] [3×3, 128]</td>
<td>×4 [3×3, 128] [3×3, 128]</td>
<td>×4 [1×1, 128] [3×3, 128]</td>
<td>×4 [1×1, 128] [3×3, 128]</td>
<td>×8 [1×1, 128] [3×3, 128]</td>
</tr>
<tr>
<td>conv4_x</td>
<td>14×14</td>
<td>×2 [3×3, 256] [3×3, 256]</td>
<td>×6 [3×3, 256] [3×3, 256]</td>
<td>×6 [1×1, 256] [3×3, 256]</td>
<td>×23 [1×1, 256] [3×3, 256]</td>
<td>×36 [1×1, 1024] [3×3, 256]</td>
</tr>
<tr>
<td>conv5_x</td>
<td>7×7</td>
<td>×2 [3×3, 512] [3×3, 512]</td>
<td>×3 [3×3, 512] [3×3, 512]</td>
<td>×3 [1×1, 512] [3×3, 512]</td>
<td>×3 [1×1, 512] [3×3, 512]</td>
<td>×3 [1×1, 2048] [3×3, 512]</td>
</tr>
<tr>
<td>1×1</td>
<td>average pool, 1000-d fc, softmax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLOPs</td>
<td>1.8×10⁹</td>
<td>3.6×10⁹</td>
<td>3.8×10⁹</td>
<td>7.6×10⁹</td>
<td>11.3×10⁹</td>
<td></td>
</tr>
</tbody>
</table>

Table from He et al. (2015)
DenseNet (2016): connect layers to all previous layers

Want to learn more?

Huang, G. et al. Densely connected convolutional networks IEEE conference on computer vision and pattern recognition (2017)

Features can incorporate global context

Figure from Hu et al. (2018)

Want to learn more?
AmoebaNet (2018): neural architecture search

Architecture found by evolution

Search acyclic graphs composed of predefined layers

Figure from Real et al. (2019)

Want to learn more?

Real, E. et al. Regularized evolution for image classifier architecture search AAAI conference on artificial intelligence (2019)
Reducing complexity

- Depthwise convolutions
- Separable convolutions
- Inverted bottlenecks (MobileNetV2, MNasNet, EfficientNet)
5

Advanced topics
Data augmentation

By design, convnets are only robust against translation.

Data augmentation makes them robust against other transformations: rotation, scaling, shearing, warping, ...
Visualising what a convnet learns

Figures from Zeiler et al. (2014)

Want to learn more?

Zeiler, M.D.; Fergus, R. 
Visualizing and understanding convolutional networks. European conference on computer vision (2014)
Visualising what a convnet learns

Figure from Simonyan et al. (2013)
Visualising what a convnet learns

Figure from Nguyen et al. (2016)
Feature Visualization

How neural networks build up their understanding of images

Feature visualization allows us to see how GoogleNet\cite{szegedy2015going}, trained on the ImageNet\cite{deng2009imagenet} dataset, builds up its understanding of images over many layers. Visualizations of all channels are available in the appendix.

AUTHORS
- Chris Olah
- Alexander Mordvintsev
- Ludwig Schubert

AFFILIATIONS
- Google Brain Team
- Google Research

PUBLISHED
- Nov. 7, 2017

DOI
- 10.23915/distill.00007
Other topics to explore

- Pre-training and fine-tuning
- Group equivariant convnets: invariance to e.g. rotation
- Recurrence and attention: other building blocks to exploit topological structure
Beyond image recognition
What else can we do with convnets?
Figures from Lin et al. (2015)
Generative models of images

- Generative adversarial nets
- Variational autoencoders
- Autoregressive models (PixelCNN)
More convnets

- Representation learning and self-supervised learning
- Convnets for video, audio, text, graphs, ...
Convolutional neural networks replaced handcrafted features with handcrafted architectures.

Prior knowledge is not obsolete: it is merely incorporated at a higher level of abstraction.
Thank you
Questions