Convolutional Models

James Allingham

University of Cambridge & Wolfram Research

26 August 2019

Goals

- Something for everyone.

Goals

- Something for everyone.
- Give a taste of techniques used in SOTA vision models.
  - Come up with your own methods!

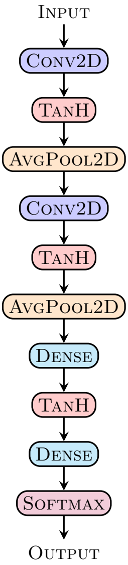
Goals

- Something for everyone.
- Give a taste of techniques used in SOTA vision models.
  - Come up with your own methods!
- Highlight some **best practises** for CNN models.

LeNet

Introduced by LeCun et al. (1998), makes use of:

- (5×5) Convolutions
- (Average) Pooling



Convolution Operation

$$
\begin{array}{ccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
*
\begin{array}{ccc}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{array}
=
\begin{array}{ccccc}
\boxed{2} & 0 & 1 & 0 & 1 \\
0 & 3 & 0 & 1 & 0 \\
0 & 0 & 3 & 0 & 0 \\
0 & 1 & 0 & 3 & 0 \\
1 & 0 & 1 & 0 & 2
\end{array}
$$

Convolution Operation

$$
\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\quad * \quad
\begin{matrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{matrix}
\quad = \quad
\begin{matrix}
2 & 0 & 1 & 0 & 1 \\
0 & 3 & 0 & 1 & 0 \\
0 & \boxed{0} & 3 & 0 & 0 \\
0 & 1 & 0 & 3 & 0 \\
1 & 0 & 1 & 0 & 2
\end{matrix}
$$

Convolution Operation

$$
\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}
\quad * \quad
\begin{matrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{matrix}
\quad = \quad
\begin{matrix}
2 & 0 & 1 & 0 & 1 \\
0 & 3 & 0 & 1 & 0 \\
0 & 0 & 3 & 0 & 0 \\
0 & \boxed{1} & 0 & 3 & 0 \\
1 & 0 & 1 & 0 & 2
\end{matrix}
$$

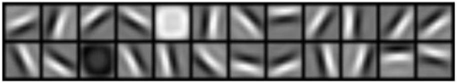
Convolution Operation

$$\begin{matrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{matrix} \quad * \quad \begin{matrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{matrix} \quad = \quad \begin{matrix} 1 & 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 1 \end{matrix}$$

Convolution Layer

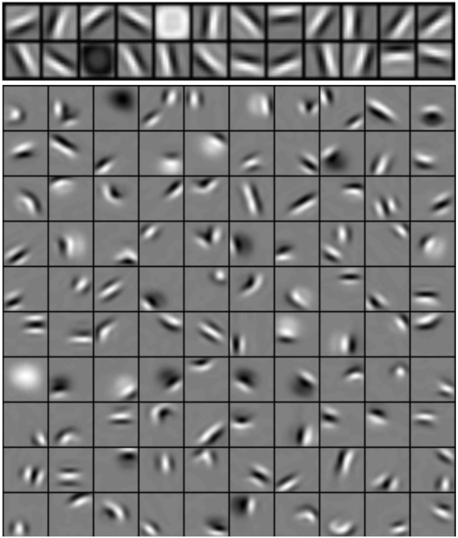
$$\textsc{Conv2D}\left(\begin{matrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{matrix}\right) \longrightarrow \begin{matrix} 1 & 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 3 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 1 \end{matrix}, \begin{matrix} 2 & 0 & 1 & 0 & 1 \\ 0 & 3 & 0 & 1 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 1 & 0 & 3 & 0 \\ 1 & 0 & 1 & 0 & 2 \end{matrix}, \dots$$

Hierarchical Features



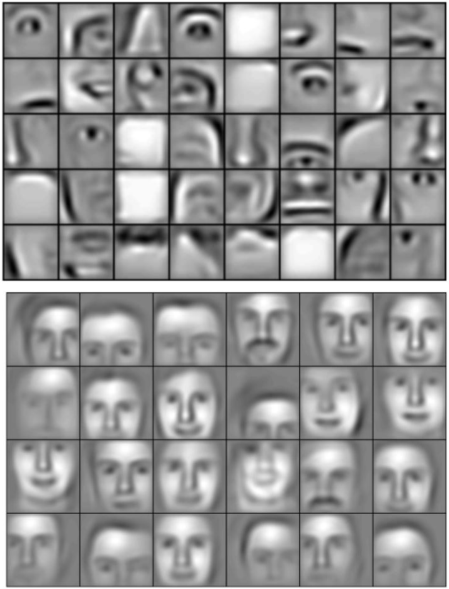
(Lee et al., 2009)

Hierarchical Features



Hierarchical Features

Hierarchical Features



(Average) Pooling Layer

$$\textsc{AvgPool2D}\left(\begin{array}{cc|cc} 9 & 1 & 0 & 2 \\ 3 & 3 & 4 & 2 \\ \hline 9 & 5 & 0 & 2 \\ 9 & 9 & 2 & 0 \end{array}\right) \longrightarrow \begin{array}{ccc} 4 & 2 & 2 \\ 5 & 3 & 2 \\ 6 & 4 & 1 \end{array}$$

(Average) Pooling Layer

$$\textsc{AvgPool2D}\left(\begin{array}{cccc} 9 & 1 & 0 & 2 \\ 3 & 3 & 4 & 2 \\ 9 & 5 & 0 & 2 \\ 9 & 9 & 2 & 0 \end{array}\right) \longrightarrow \begin{array}{cc} 4 & 2 \\ 6 & 1 \end{array}$$

Receptive Fields

3×3 Conv

9	1	0	2	5
3	3	4	2	0
9	5	0	2	0
9	9	2	6	1
5	4	2	9	2

4	2	0
6	1	0
9	5	5

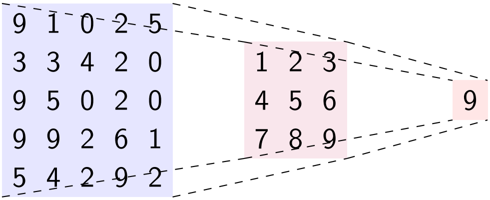
Receptive Fields

4×4 CONV

$$\begin{matrix} 9 & 1 & 0 & 2 & 5 \\ 3 & 3 & 4 & 2 & 0 \\ 9 & 5 & 0 & 2 & 0 \\ 9 & 9 & 2 & 6 & 1 \\ 5 & 4 & 2 & 9 & 2 \end{matrix} \qquad \begin{matrix} \boxed{4} & 2 & 0 \\ 6 & 1 & 0 \\ 9 & 5 & 5 \end{matrix}$$

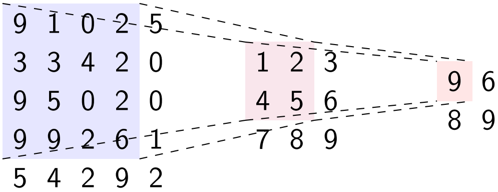
Receptive Fields

$3{\times}3$ CONV $\rightarrow$ $3{\times}3$ CONV

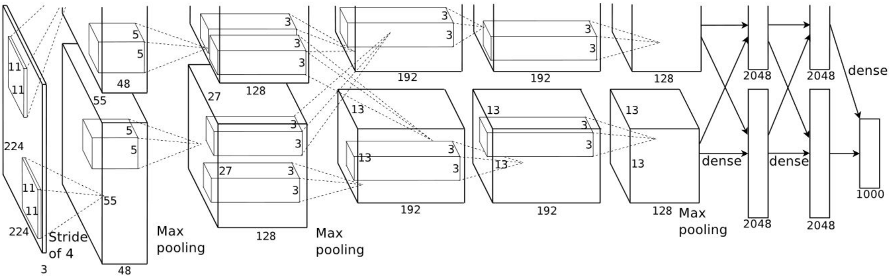


Receptive Fields

$3{\times}3$ CONV $\rightarrow$ $2{\times}2$ POOL



AlexNet

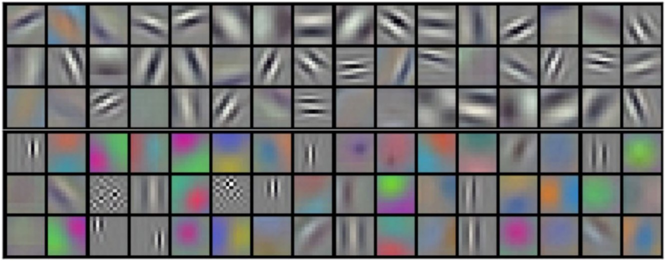


Introduced by Krizhevsky et al. (2012), makes use of:

- Grouped convolutions (various sizes)
- (Overlapping) **Max** pooling
- **ReLU** non-linearity
- Local response norm
- **Dropout**

AlexNet

Learned Features



(Max) Pooling Layer

$$\textsc{MaxPool2D}\begin{pmatrix} 9 & 1 & 0 & 2 \\ 3 & 3 & 4 & 2 \\ 9 & 5 & 0 & 2 \\ 9 & 9 & 2 & 0 \end{pmatrix} \longrightarrow \begin{matrix} 9 & 4 & 4 \\ 9 & 5 & 4 \\ 9 & 9 & 2 \end{matrix}$$

ReLU Activation Layer

$$\textsc{ReLU}\begin{pmatrix} 2 & -1 & 0 & 2 \\ 1 & 3 & -4 & -2 \\ 4 & 5 & 0 & 2 \\ -2 & -8 & 0 & -3 \end{pmatrix} \longrightarrow \begin{matrix} 2 & 0 & 0 & 2 \\ 1 & 3 & 0 & 0 \\ 4 & 5 & 0 & 2 \\ 0 & 0 & 0 & 0 \end{matrix}$$

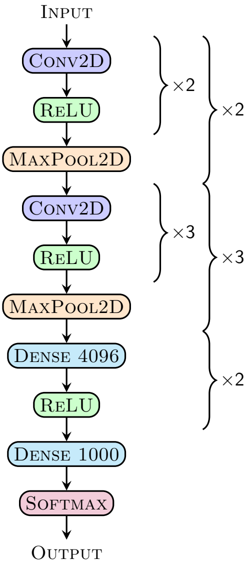
Dropout Layer

$$\textsc{Dropout}\begin{pmatrix} 7 & 2 & 2 & 1 \\ 3 & 1 & 8 & 4 \\ 2 & 6 & 4 & 2 \\ 3 & 3 & 5 & 1 \end{pmatrix} \longrightarrow \begin{array}{cccc} 0 & 4 & 4 & 2 \\ 6 & 0 & 0 & 0 \\ 4 & 0 & 8 & 0 \\ 6 & 6 & 0 & 0 \end{array}$$

VGG

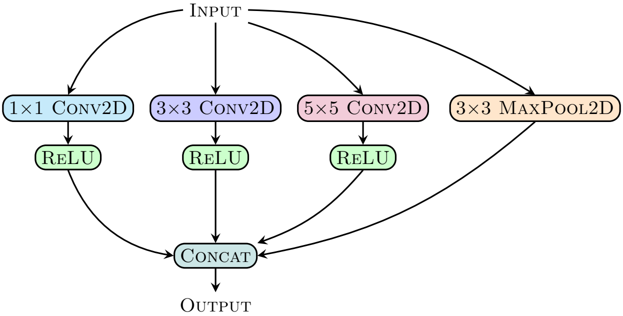
Introduced by Simonyan and Zisserman (2014).

- Only **3×3 Convolutions**
- Only **2×2 Max Pooling**



Inception V1

AKA GoogLeNet

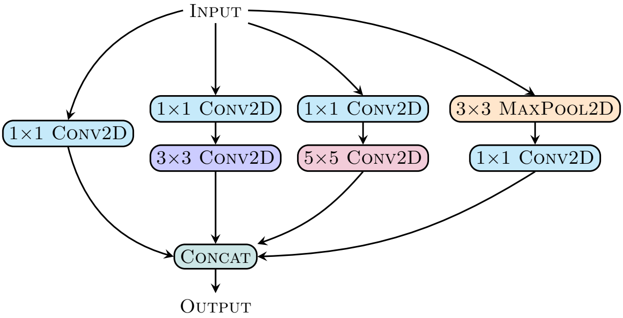


Introduced by Szegedy et al. (2014).

- Go a bit wider rather than deeper (still 27 layers).
  - With *Inception Modules* (9 of them).
- Convolutions of different sizes make a come back!
- Including **1×1 Convolutions**??? (Lin et al., 2013)

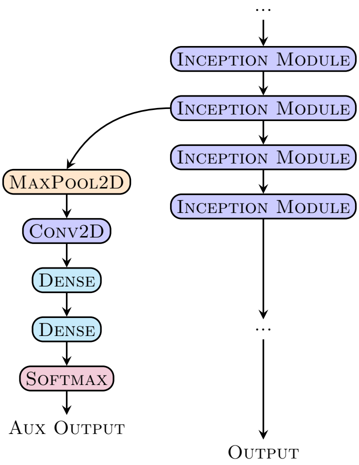
Inception V1

AKA GoogLeNet



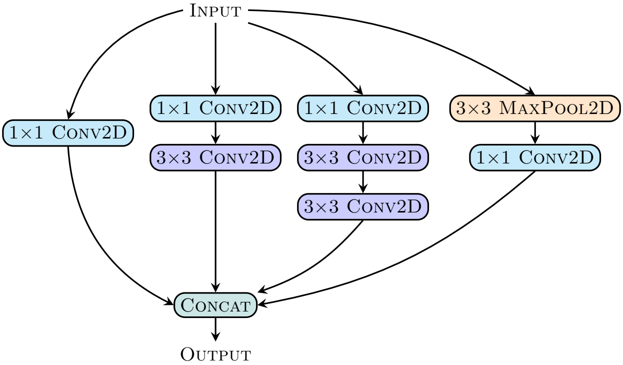
Inception V1

Auxiliary Classifier – Vanishing Gradients

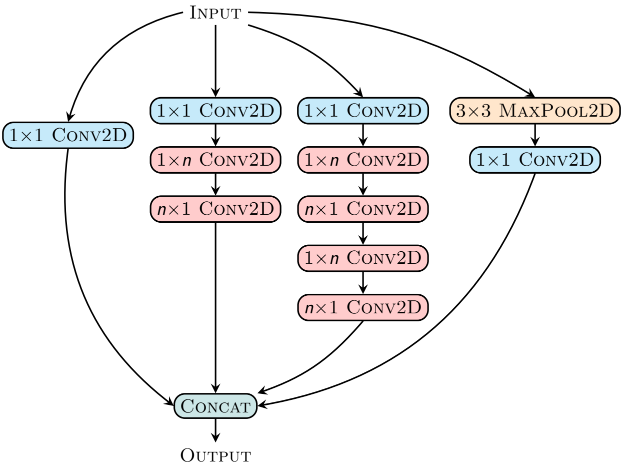


Inception V2

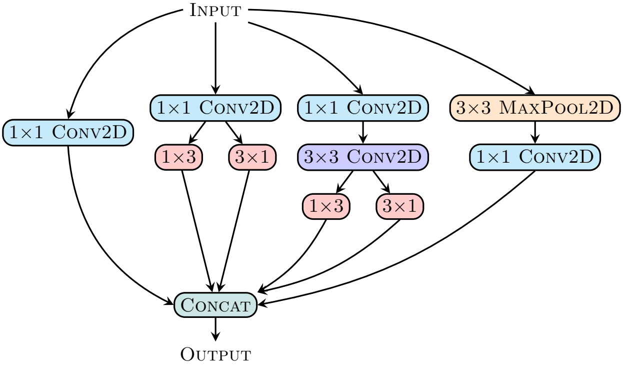
Introduced by Szegedy et al. (2015).



Inception V2



Inception V2



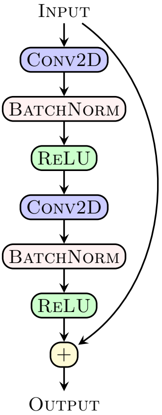
Inception V3

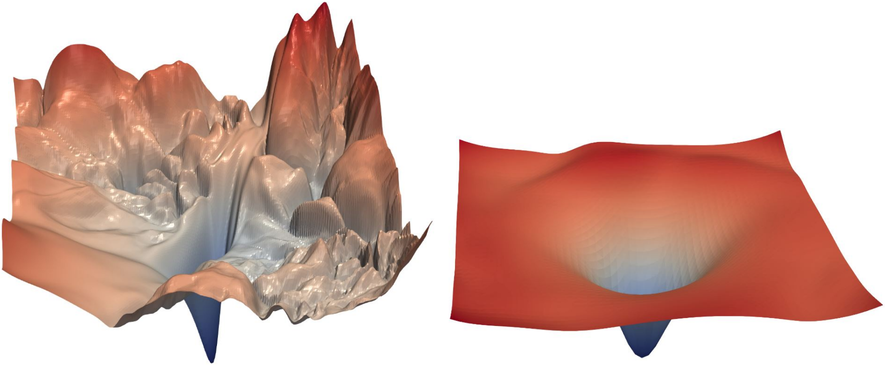
Also introduced by Szegedy et al. (2015).

- 7×7 Convolutions make a comeback!
- Various training improvements.
  - **Batch normalisation**.
  - Label smoothing.
  - RMSProp.

Introduced by He et al. (2015).

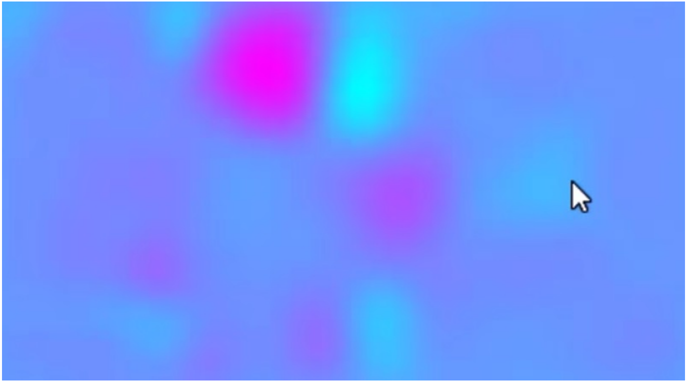
- **Residual connections**.
  - Bye-bye vanishing gradients.
  - Much deeper (100s of layers)!
- Fully-Convolutional
  - Dense $\rightarrow$ global average pooling.
  - Less over-fitting.
  - Heat-maps!
- Only 3×3 convolutions.
- **Little max pooling**.





(Li et al., 2017)

(Adapted from FastAI's Practical Deep Learning for Coders 2017)



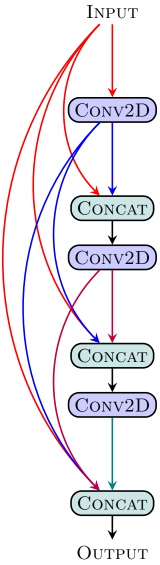
(Adapted from FastAI's Practical Deep Learning for Coders 2017)

(Adapted from FastAI's Practical Deep Learning for Coders 2017)

DenseNet

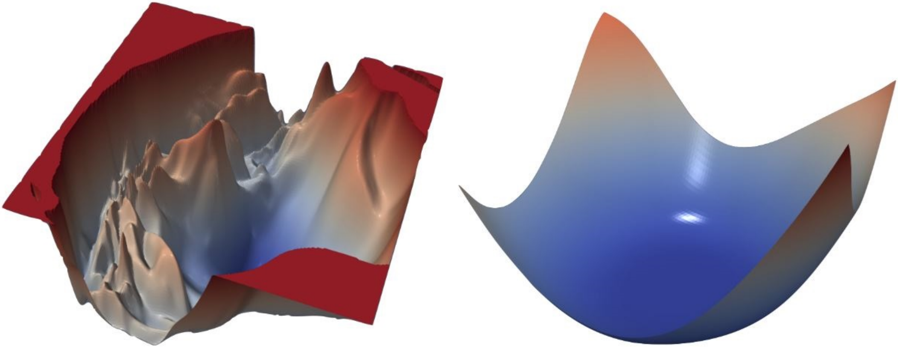
Introduced by Huang et al. (2016).

- **Dense connections**.
- 121 layers (but more like 10).
- 1×1 convolutions as *bottleneck* layers before expensive 3×3 convolutions.



DenseNet

What the skip connection does

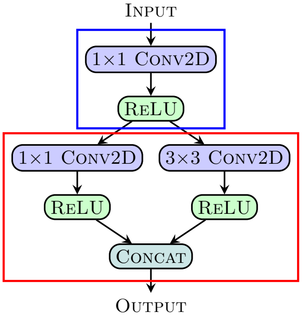


(Li et al., 2017)

SqueezeNet

Introduced by Iandola et al. (2016).

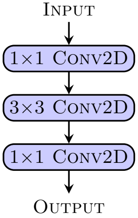
- $3 \times 3 \rightarrow 1 \times 1$ convolutions.
- Reduce number of channels.
- Downsample later in the net.
- Fire module
  - Squeeze and Expansion layers.
- Same accuracy as AlexNet but *50*$\times$ fewer weights.
  - No dense layers.
  - $<$ 0.5MB model size.



MobileNet

Introduced by Howard et al. (2017).

- Depthwise separable convolutions.
- Very flexible *family* of nets.
- Also fully-convolutional.



What I didn't talk about...

- Anything other than image classification!

What I didn't talk about...

- Anything other than image classification!
  - But don't worry, a lot of this applies to other tasks.

What I didn't talk about...

- Anything other than image classification!
  - But don't worry, a lot of this applies to other tasks.
- **YOLO** for **Object Detection** (Redmon et al., 2015).
  - Very similar structure to VGG but with auxiliary outputs.

What I didn't talk about...

- Anything other than image classification!
  - But don't worry, a lot of this applies to other tasks.
- **YOLO** for **Object Detection** (Redmon et al., 2015).
  - Very similar structure to VGG but with auxiliary outputs.
- **100 Layers Tiramisu** and **UNet** for **Image Segmentation** (Jégou et al., 2016; Ronneberger et al., 2015).
  - Based on DenseNets and ResNets.

What I didn't talk about...

- Anything other than image classification!
  - But don't worry, a lot of this applies to other tasks.
- **YOLO** for **Object Detection** (Redmon et al., 2015).
  - Very similar structure to VGG but with auxiliary outputs.
- **100 Layers Tiramisu** and **UNet** for **Image Segmentation** (Jégou et al., 2016; Ronneberger et al., 2015).
  - Based on DenseNets and ResNets.
- Deconvolutions, Upsampling, GANs, etc. for **Image Synthesis** and **Super Resolution**.
  - e.g. ESRGAN (Wang et al., 2018).

What I didn't talk about...

- Anything other than image classification!
  - But don't worry, a lot of this applies to other tasks.
- **YOLO** for **Object Detection** (Redmon et al., 2015).
  - Very similar structure to VGG but with auxiliary outputs.
- **100 Layers Tiramisu** and **UNet** for **Image Segmentation** (Jégou et al., 2016; Ronneberger et al., 2015).
  - Based on DenseNets and ResNets.
- Deconvolutions, Upsampling, GANs, etc. for **Image Synthesis** and **Super Resolution**.
  - e.g. ESRGAN (Wang et al., 2018).
- Convolutions for language models!
  - e.g. Conv Seq2Seq (Gehring et al., 2017).

What I didn't talk about...

- Anything other than image classification!
  - But don't worry, a lot of this applies to other tasks.
- **YOLO** for **Object Detection** (Redmon et al., 2015).
  - Very similar structure to VGG but with auxiliary outputs.
- **100 Layers Tiramisu** and **UNet** for **Image Segmentation** (Jégou et al., 2016; Ronneberger et al., 2015).
  - Based on DenseNets and ResNets.
- Deconvolutions, Upsampling, GANs, etc. for **Image Synthesis** and **Super Resolution**.
  - e.g. ESRGAN (Wang et al., 2018).
- Convolutions for language models!
  - e.g. Conv Seq2Seq (Gehring et al., 2017).

Thank You!

References I

GEHRING, Jonas, AULI, Michael, GRANGIER, David, YARATS, Denis and DAUPHIN, Yann N (2017). Convolutional Sequence to Sequence Learning. *ArXiv e-prints*. 1705.03122.

HE, Kaiming, ZHANG, Xiangyu, REN, Shaoqing and SUN, Jian (2015). Deep residual learning for image recognition. *CoRR*, **abs/1512.03385**. 1512.03385, URL http://arxiv.org/abs/1512.03385.

HOWARD, Andrew G., ZHU, Menglong, CHEN, Bo, KALENICHENKO, Dmitry, WANG, Weijun, WEYAND, Tobias, ANDREETTO, Marco and ADAM, Hartwig (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, **abs/1704.04861**. 1704.04861, URL http://arxiv.org/abs/1704.04861.

HUANG, Gao, LIU, Zhuang and WEINBERGER, Kilian Q. (2016). Densely connected convolutional networks. *CoRR*, **abs/1608.06993**. 1608.06993, URL http://arxiv.org/abs/1608.06993.

References II

Iandola, Forrest N., Moskewicz, Matthew W., Ashraf, Khalid, Han, Song, Dally, William J. and Keutzer, Kurt (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, **abs/1602.07360**. 1602.07360, URL http://arxiv.org/abs/1602.07360.

Jégou, Simon, Drozdzal, Michal, Vázquez, David, Romero, Adriana and Bengio, Yoshua (2016). The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. *CoRR*, **abs/1611.09326**. 1611.09326, URL http://arxiv.org/abs/1611.09326.

Krizhevsky, Alex, Sutskever, Ilya and Hinton, Geoffrey E (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

References III

LeCun, Yann, Bottou, Leon, Bengio, Y and Haffner, Patrick (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86** 2278–2324.

Lee, Honglak, Grosse, Roger, Ranganath, Rajesh and Ng, Andrew Y (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th annual international conference on machine learning*, 609–616. ACM.

Li, Hao, Xu, Zheng, Taylor, Gavin and Goldstein, Tom (2017). Visualizing the loss landscape of neural nets. *CoRR*, **abs/1712.09913**. 1712.09913, URL http://arxiv.org/abs/1712.09913.

Lin, Min, Chen, Qiang and Yan, Shuicheng (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

References IV

Redmon, Joseph, Divvala, Santosh Kumar, Girshick, Ross B. and Farhadi, Ali (2015). You only look once: Unified, real-time object detection. *CoRR*, **abs/1506.02640**. 1506.02640, URL http://arxiv.org/abs/1506.02640.

Ronneberger, Olaf, Fischer, Philipp and Brox, Thomas (2015). U-net: Convolutional networks for biomedical image segmentation. *CoRR*, **abs/1505.04597**. 1505.04597, URL http://arxiv.org/abs/1505.04597.

Simonyan, Karen and Zisserman, Andrew (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

References V

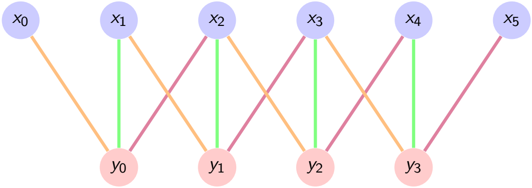
SZEGEDY, Christian, LIU, Wei, JIA, Yangqing, SERMANET, Pierre, REED, Scott, ANGUELOV, Dragomir, ERHAN, Dumitru, VANHOUCKE, Vincent and RABINOVICH, Andrew (2014). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

SZEGEDY, Christian, VANHOUCKE, Vincent, IOFFE, Sergey, SHLENS, Jonathon and WOJNA, Zbigniew (2015). Rethinking the inception architecture for computer vision. *CoRR*, **abs/1512.00567**. `1512.00567`, URL `http://arxiv.org/abs/1512.00567`.

WANG, Xintao, YU, Ke, WU, Shixiang, GU, Jinjin, LIU, Yihao, DONG, Chao, LOY, Chen Change, QIAO, Yu and TANG, Xiaoou (2018). ESRGAN: enhanced super-resolution generative adversarial networks. *CoRR*, **abs/1809.00219**. `1809.00219`, URL `http://arxiv.org/abs/1809.00219`.

Parameter Sharing

Convolution Layer



Fully-connected Layer