Neural Networks, Deep Learning

Jose Martinez Heras

22/03/2018

Resources

Watch the video of this lecture

https://dlmultimedia.esa.int/download/public/videos/2048/03/008/4803_008_AR_EN.mp4

Watch the practical exercise video

https://dlmultimedia.esa.int/download/public/videos/2048/03/007/4803_007_AR_EN.mp4

Get presentation and additional resources on

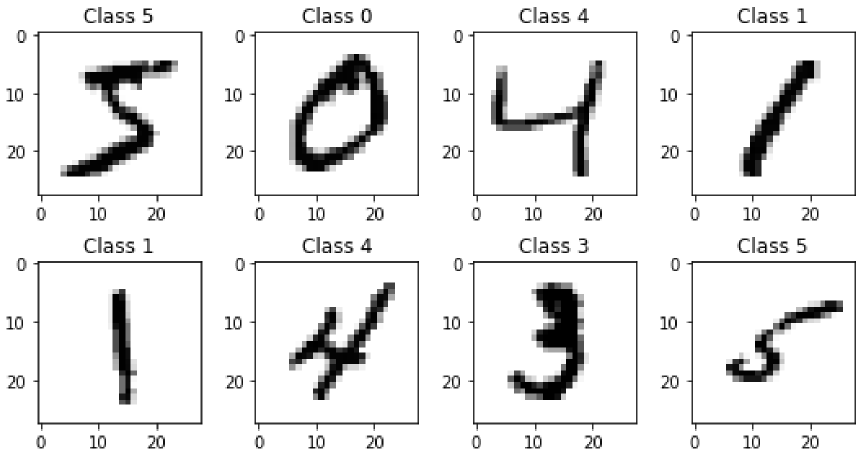
https://github.com/jmartinezheras/2018-MachineLearning-Lectures-ESA

Outline for Supervised Learning (3)

Session 4: Supervised Learning (3)

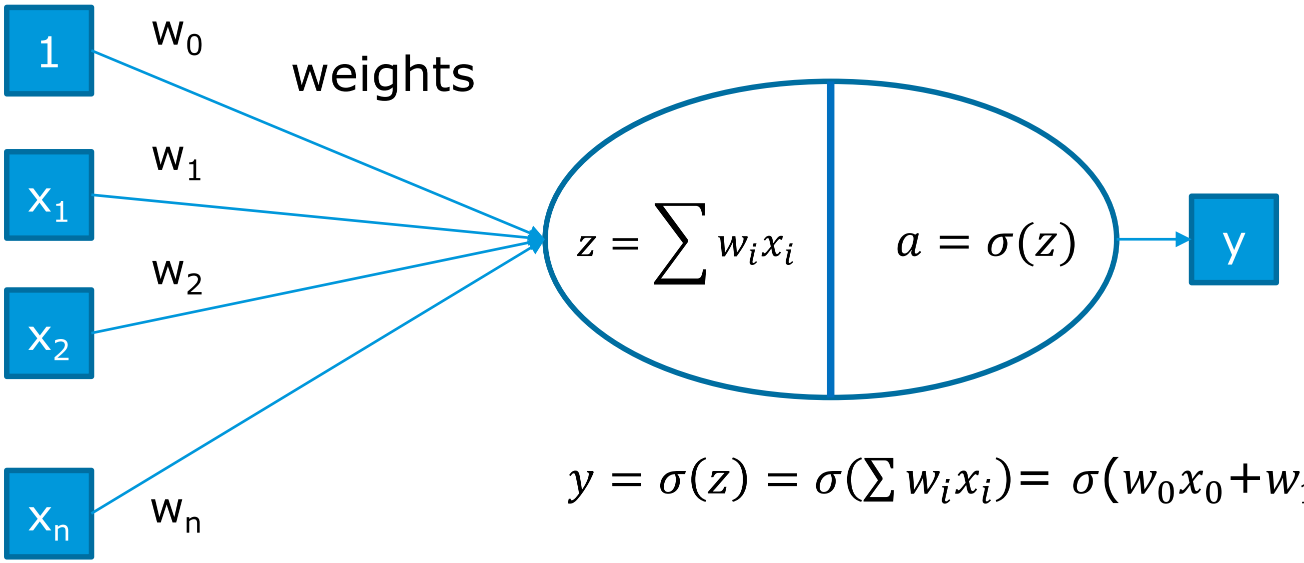
- Neural Networks
- Deep Learning
  - Convolutional Networks
  - Recurrent Networks

- Hands-on: image classification



Neural Networks

Let’s create a simple neural network with 1 neuron

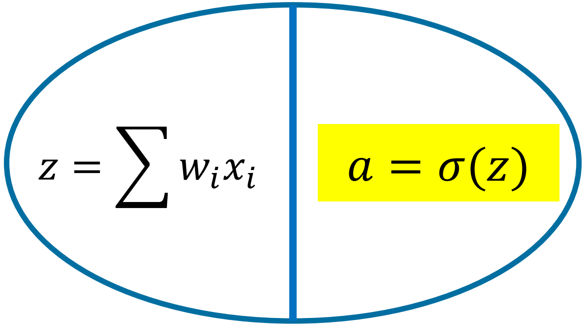


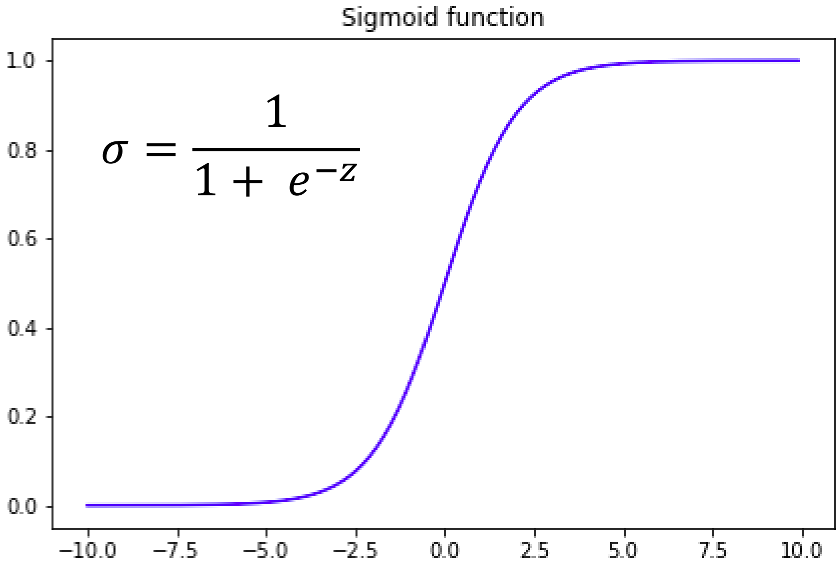
$$y = \sigma(z) = \sigma(\sum w_i x_i) = \sigma(w_0 x_0 + w_1 x_1 + \ldots w_n x_n)$$

Sigmoid Activation Function

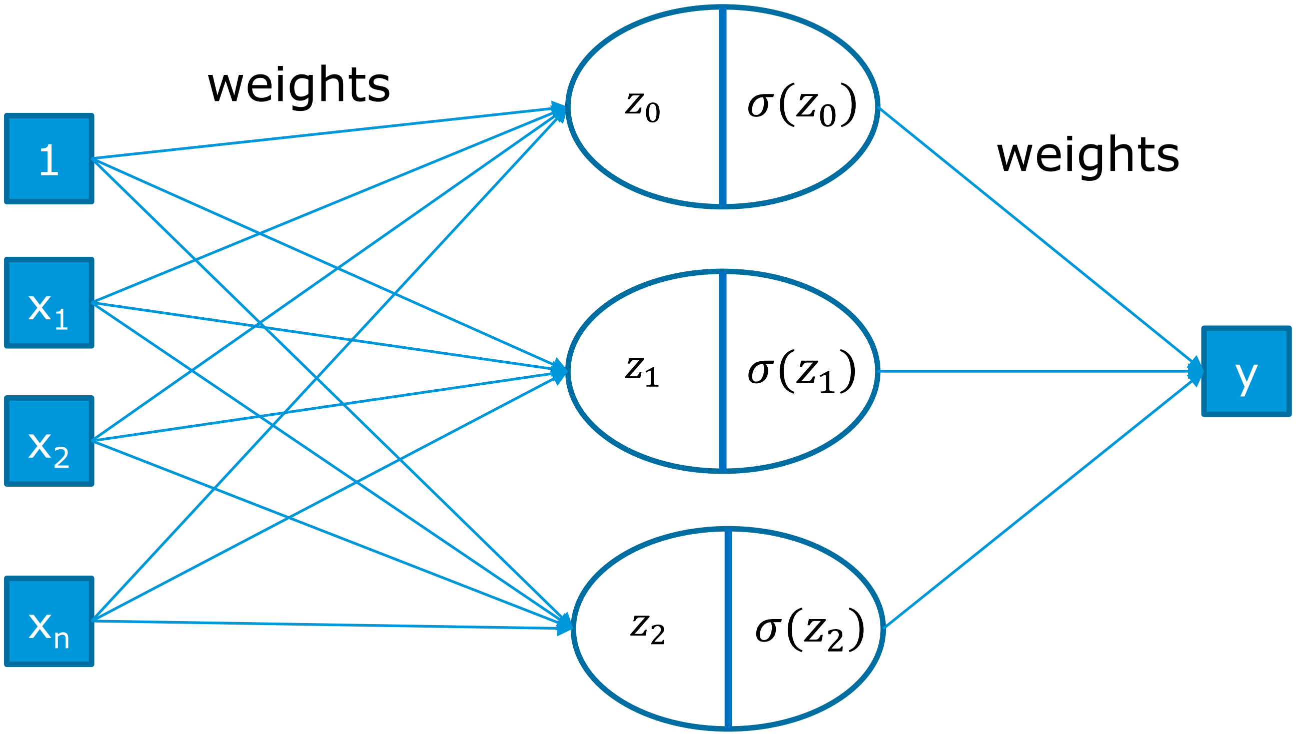
Sigmoid Function

- Values [0, 1]

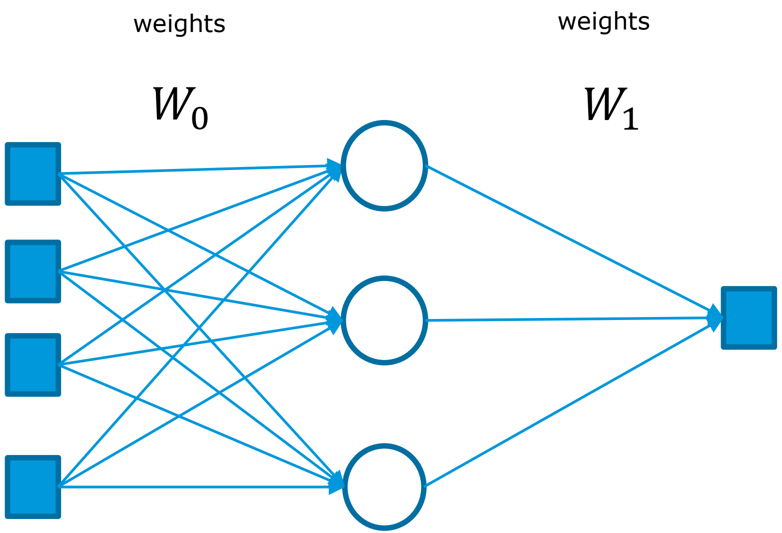




Neural Networks



How Neural Networks learn? – Back Propagation



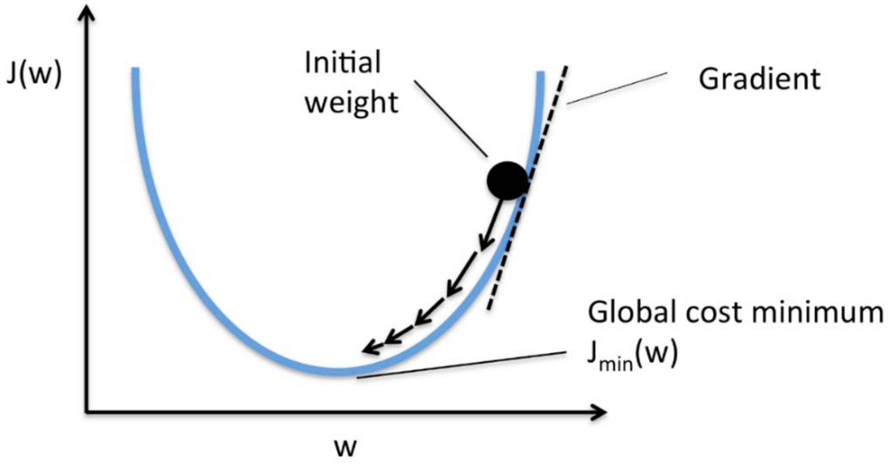
Learning previous layers

Use chain rule to compute the previous layers

Learning last layer

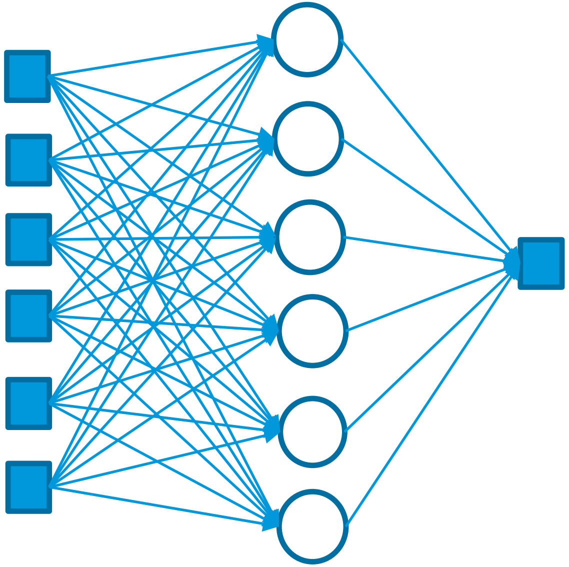
$$W_1 = W_1 - \alpha \frac{\partial J}{\partial W_1}$$

$$J = MSE = \frac{1}{m}\sum^{m}(Y_i - \widehat{Y}_i)^2$$

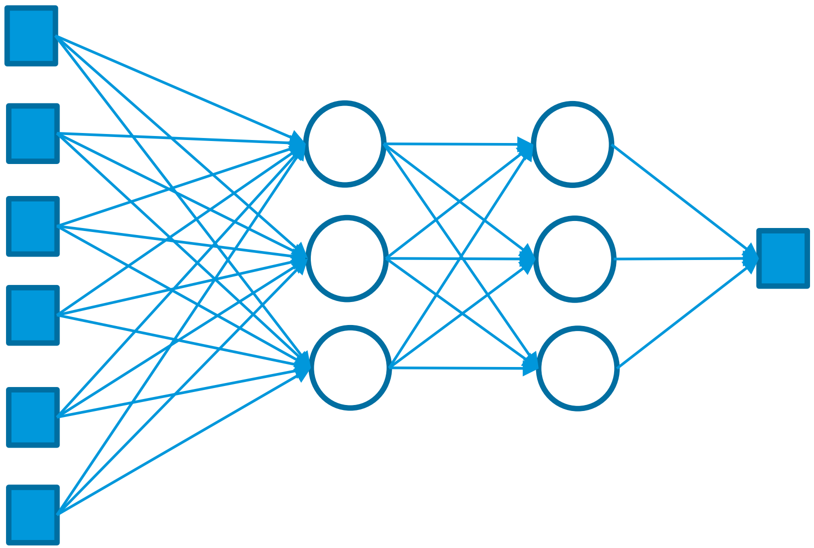


Gradient Descent Visualization. Credit: rasbt.github.io

What about Deep Learning?



Shallow Network



Deep Network

Deep Learning scales better with the amount of data

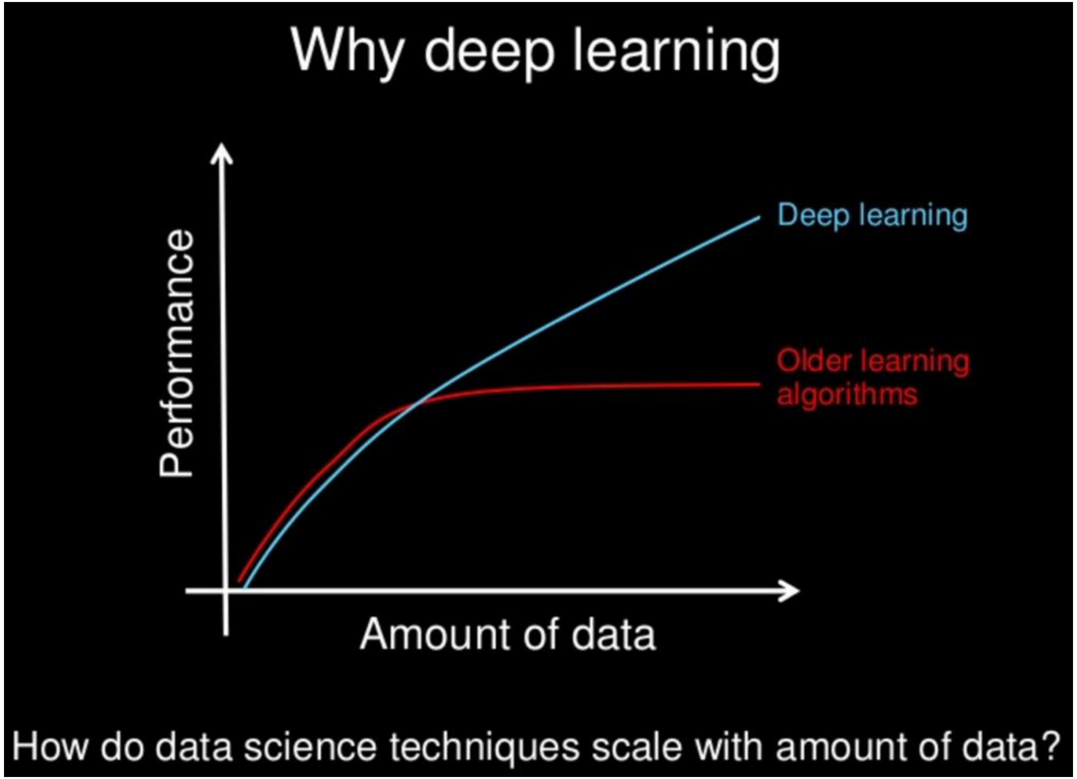


Image credit: https://www.slideshare.net/ExtractConf

Representation Learning

Deep Networks can learn more complex patterns because they find the hierarchies present in the data

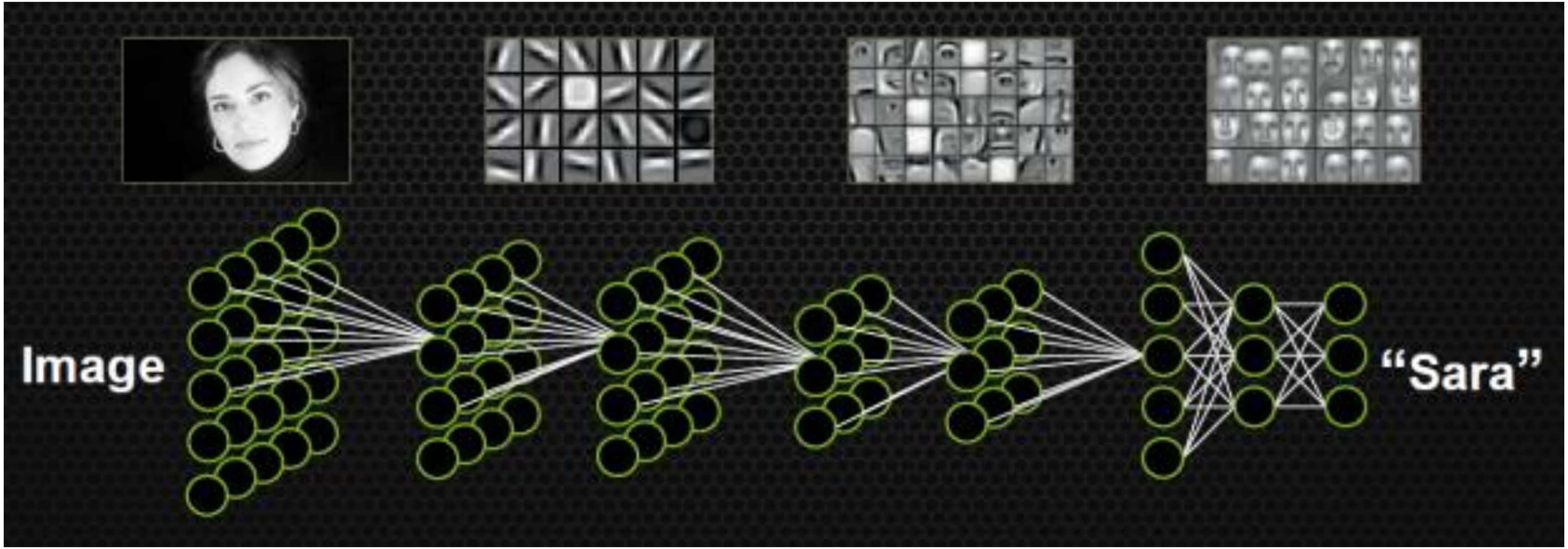
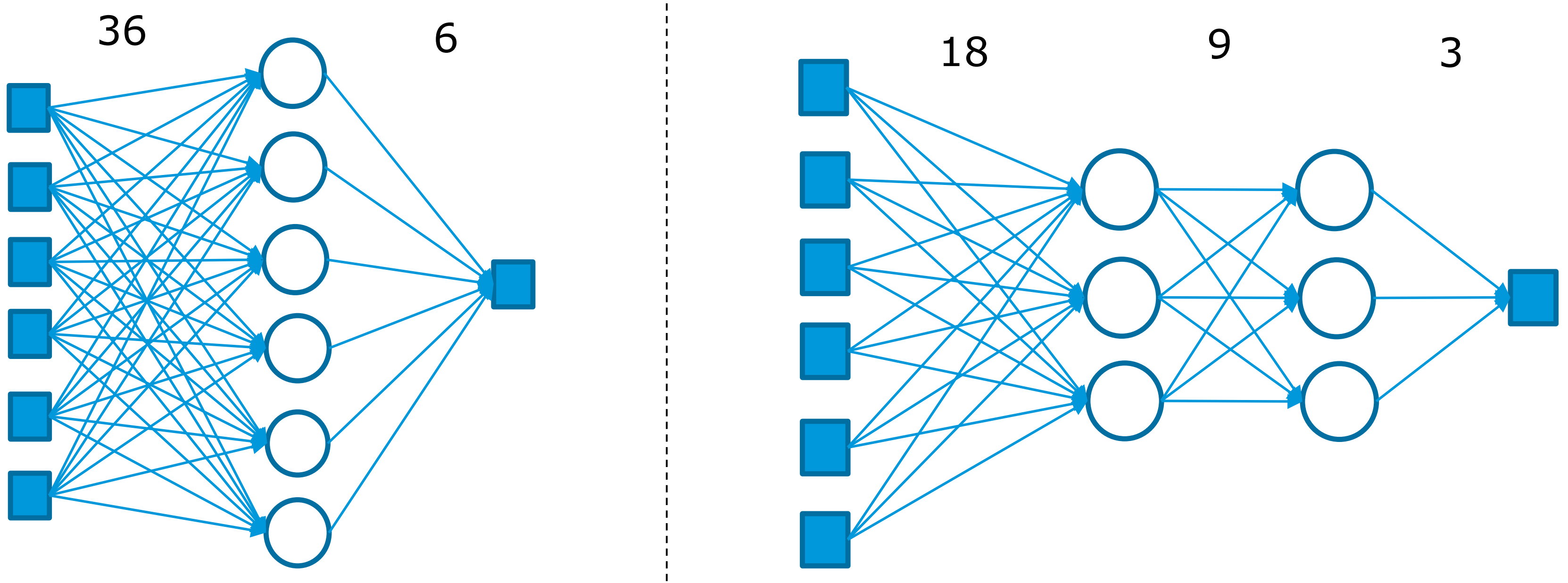


Image credit: https://devblogs.nvidia.com/accelerate-machine-learning-cudnn-deep-neural-network-library/

Deeper Network need less parameters for the same task

In case a shallow network can perform the same task, the deep network often perform better with less parameters



Shallow Network: 40 parameters

Deep Network: 30 parameters

Deep Learning

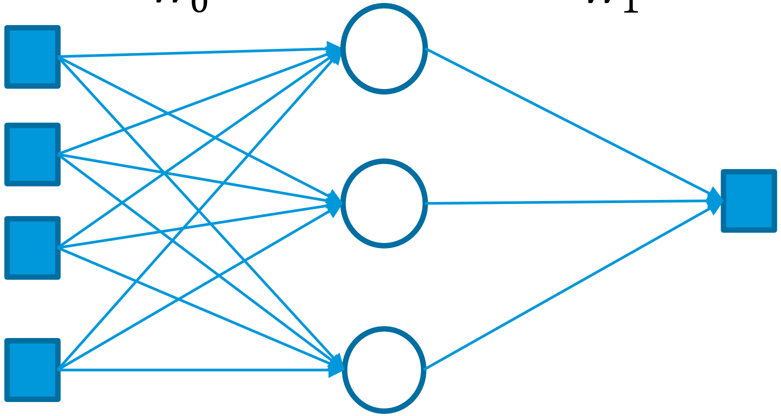
So, deep learning is the way to go

Neural Networks have been around for many years…

Why it took so long for Deep Learning?

- More data available
- More processing power: Graphical Processing Units (GPUs)
- **Vanishing gradient**

How Neural Networks learn? – Back Propagation



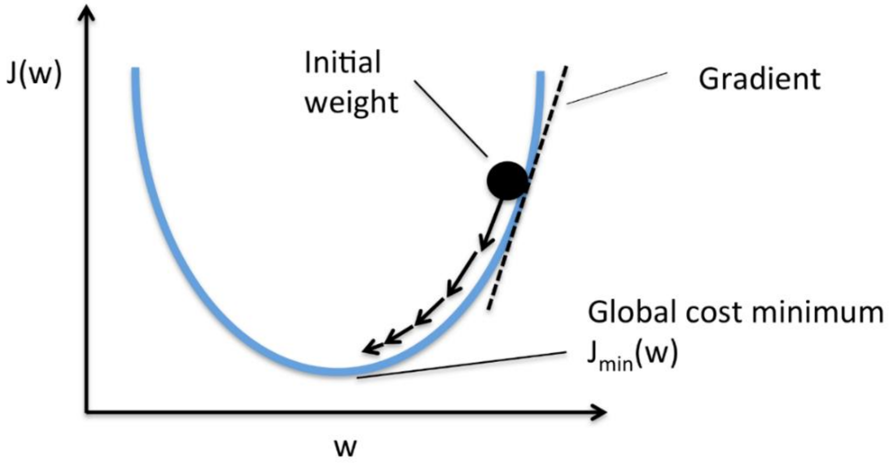
Learning previous layers

Use chain rule to compute the previous layers

Learning last layer

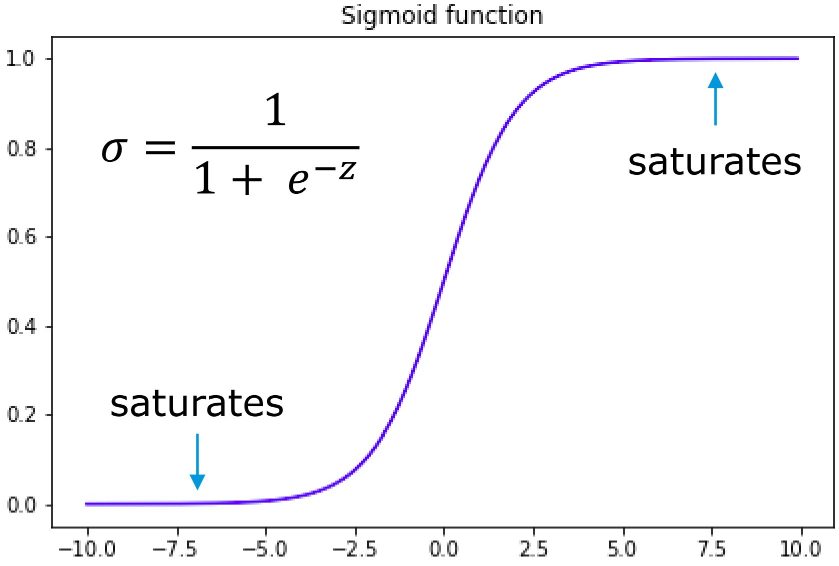
$$W_1 = W_1 - \alpha \frac{\partial J}{\partial W_1}$$

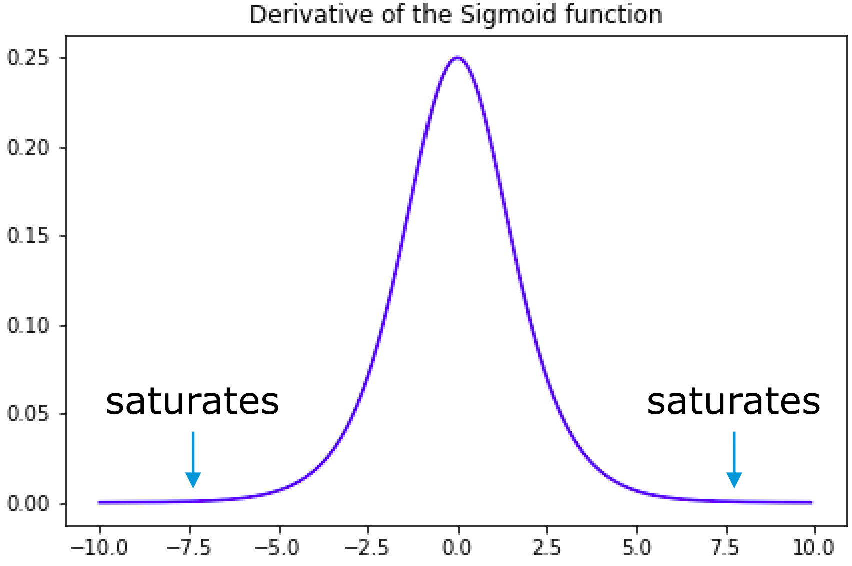
$$J = MSE = \frac{1}{m}\sum^{m}(Y_i - \widehat{Y}_i)^2$$



Gradient Descent Visualization. Credit: rasbt.github.io

Vanishing Gradient





$$\sigma'(z) = \sigma(z) \cdot (1 - \sigma(z))$$

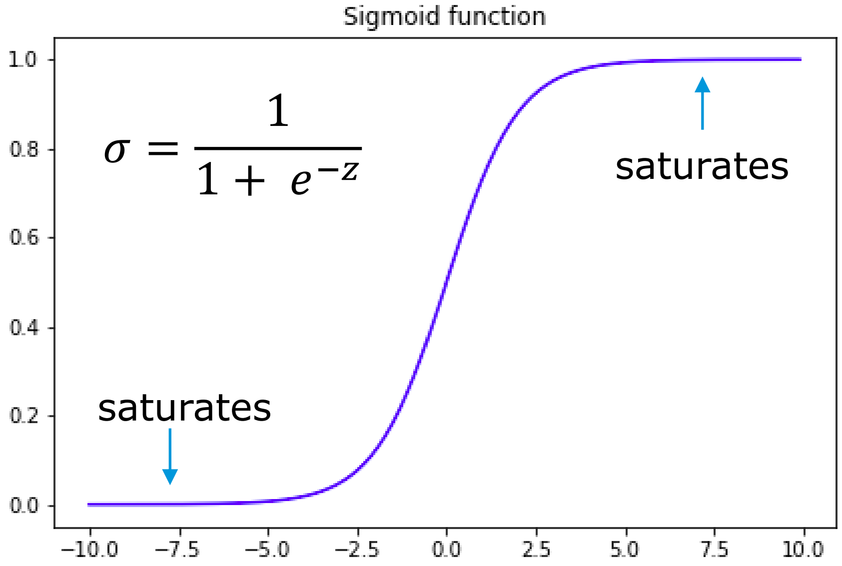
Vanishing Gradient – Better weights initialization

Better initialization of weights

Default weight initialization before 2010

$W_i \sim \mathrm{N}(0,1)$

caused that the activation of the sigmoid function was close to saturation points



Xavier initialization

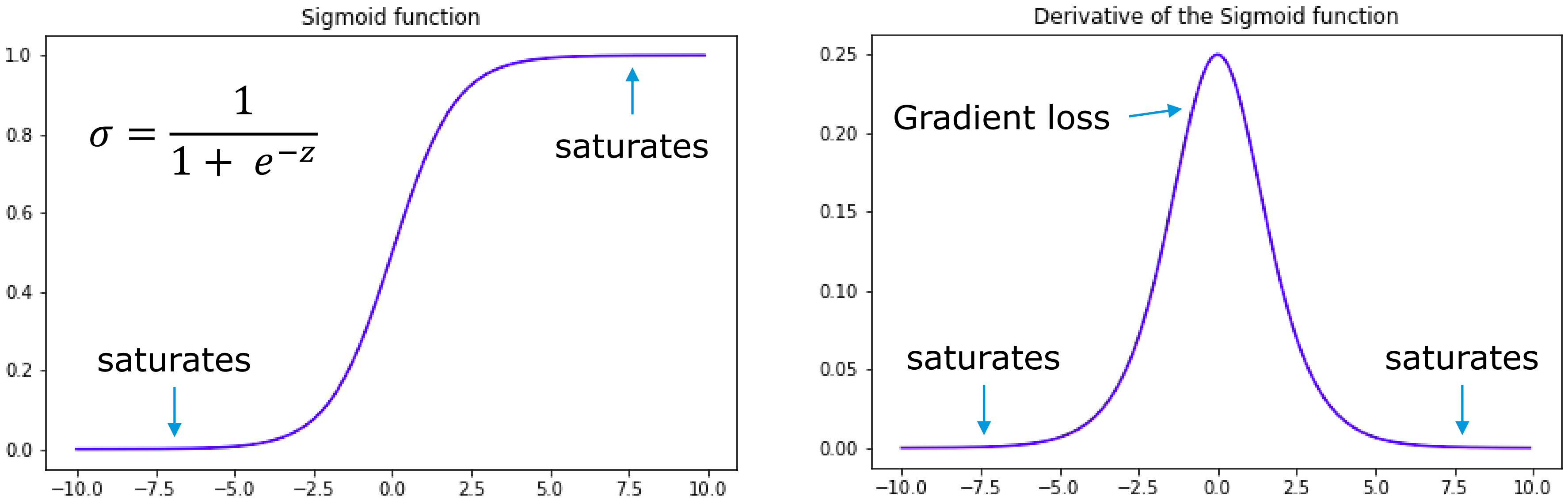
$W_i \sim \mathrm{N}(0, \mathrm{std})$ note: a different std per neuron

$$std = \sqrt{\frac{2}{n_{inputs} + n_{outputs}}}$$

$n_{inputs} = number\ of\ inputs\ to\ the\ neuron$

$n_{outputs} = number\ of\ outputs\ from\ the\ neuron$

Vanishing Gradient – Even with better weights



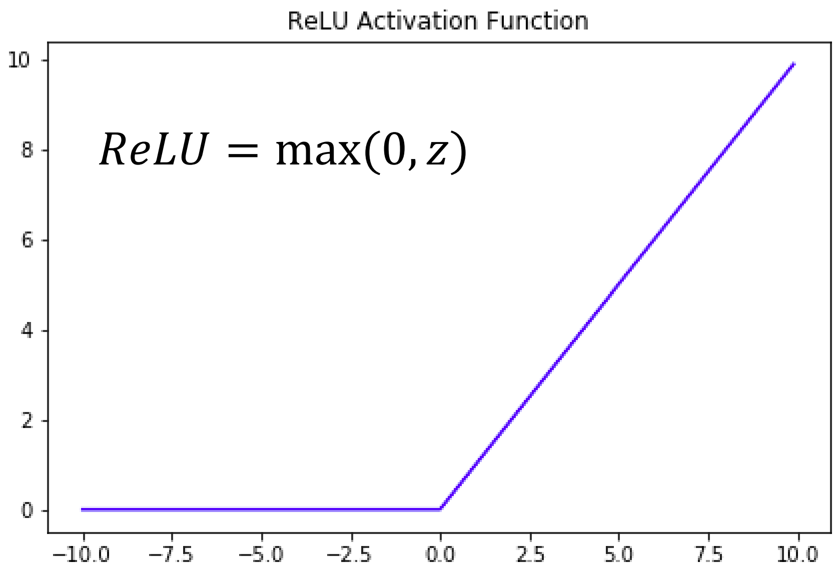
Even with better initialization of weights, the sigmoid can only pass to the previous layer a ¼ of the gradient (in the best case)

Vanishing Gradient - solutions

Better activation functions

Before 2010 most people were using sigmoid functions because neuroscientists discovered that biological neurons worked this way …

- **Rectified Linear Unit (ReLU)**
- Leaky ReLu
- Exponential Linear Unit (ELU)

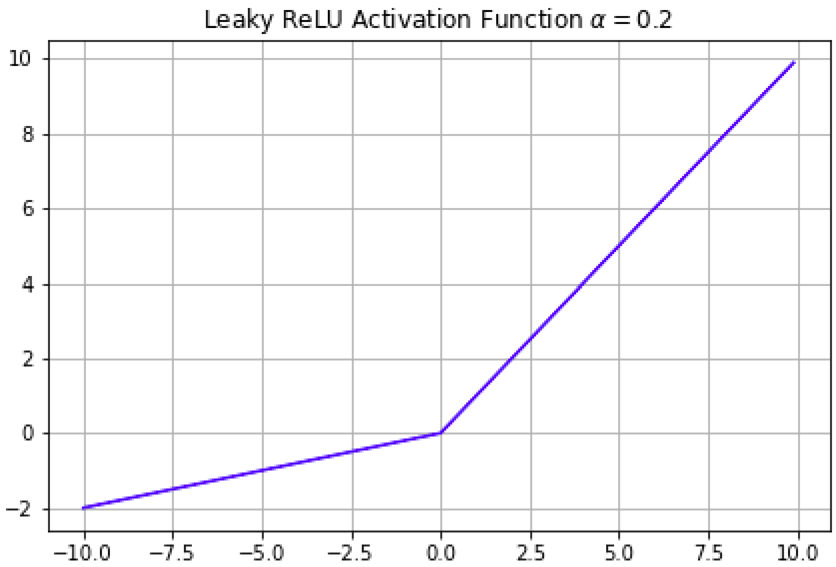


Vanishing Gradient - solutions

Better activation functions

Before 2010 most people were using sigmoid functions because neuroscientists discovered that neurons worked this way …

- Rectified Linear Unit (ReLU)
- **Leaky ReLu**
- Exponential Linear Unit (ELU)



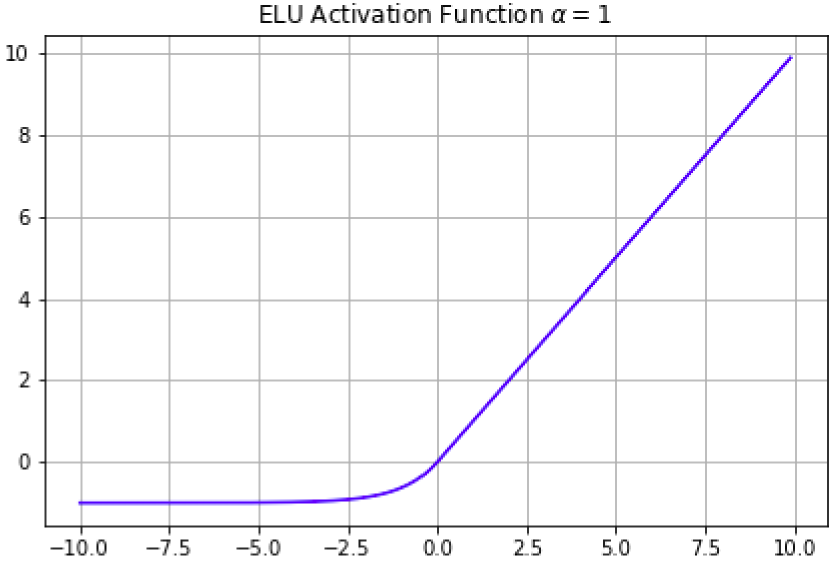
$$Leaky\ ReLU = \max(\alpha \cdot z, z)$$

Vanishing Gradient - solutions

Better activation functions

Before 2010 most people were using sigmoid functions because neuroscientists discovered that neurons worked this way …

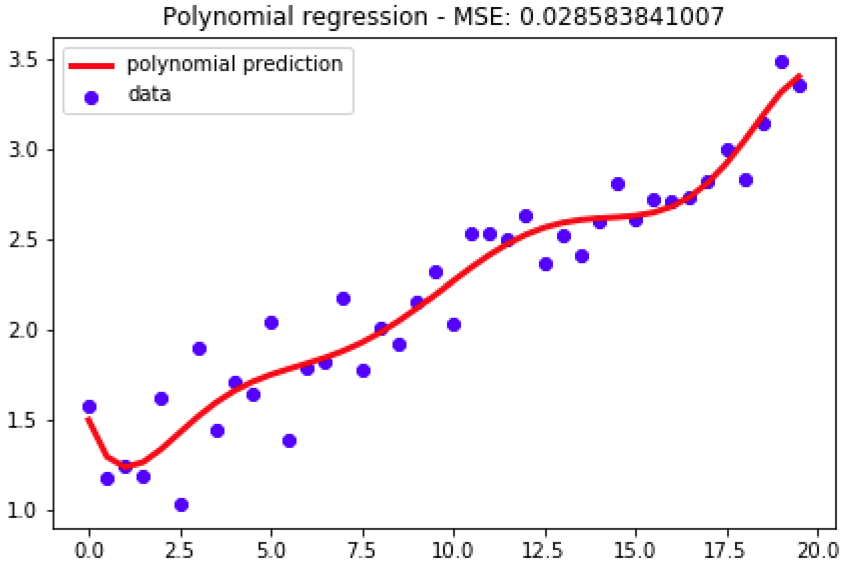
- Rectified Linear Unit (ReLU)
- Leaky ReLu
- **Exponential Linear Unit (ELU)**

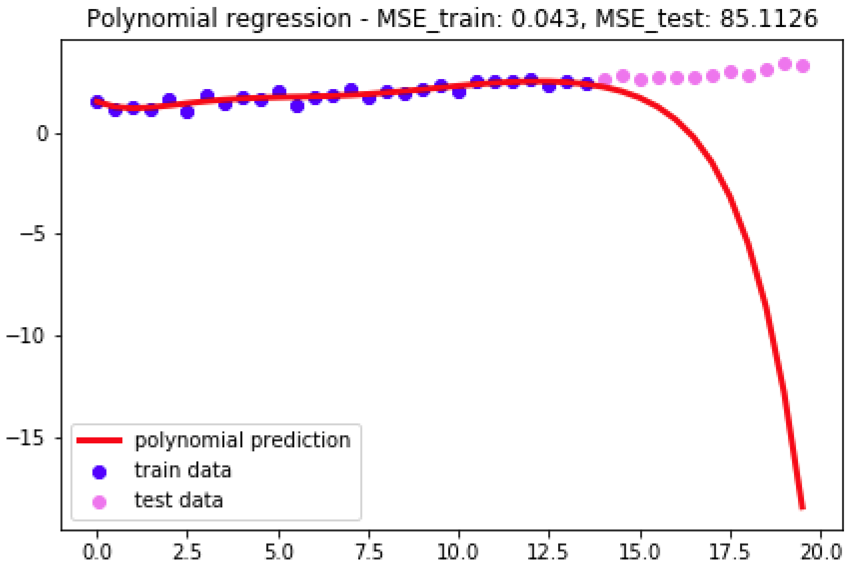


$$ELU = \begin{cases} z \; if \; z \geq 0 \\ \alpha \cdot (e^z - 1) \; if \; z < 0 \end{cases}$$

Generalization

Generalization: the ability of a model to perform well on new data





Generalization

To measure generalization:

- Train data: to fit the model to the data
- Test data: to evaluate how the model generalizes
- Validation (development) data: to select which model, hyper-pameters

Classic Machine Learning (≤ 10,000 samples)

Train – 60%	Val – 20%	Test – 20%

Deep Learning (> 1,000,000 samples)

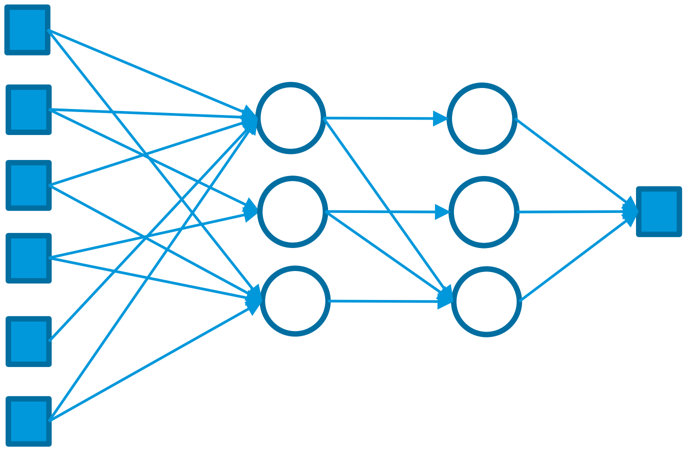
Train – 98%	Val – 1%	Test – 1%

Deep Learning - Regularization

- L1 (Lasso) $$J = MSE + \alpha \sum_{j=1}^{n} |w_j|$$

- L2 (Ridge) $$J = MSE + \alpha \frac{1}{2} \sum_{j=1}^{n} w_j^2$$

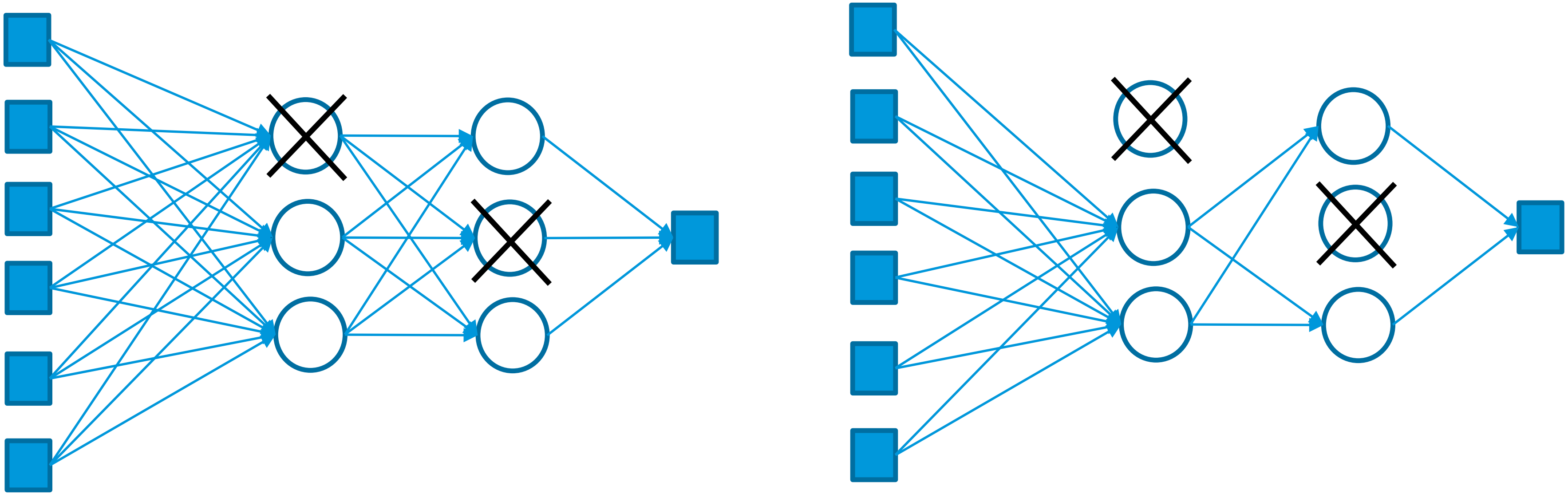
- ElasticNet $$J = MSE + r \cdot Lasso + (1 - r) \cdot Ridge$$



Deep Learning – Regularization - Dropout

Dropout is the most effective regularization technique (2012)

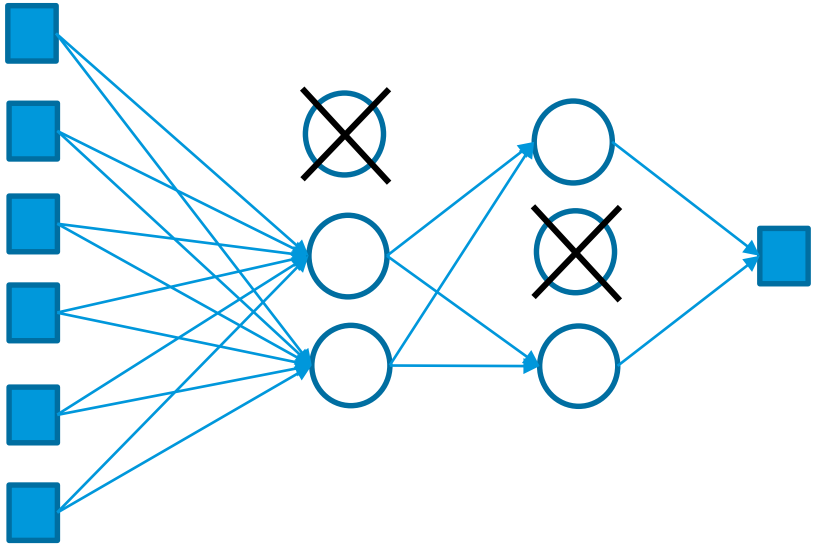
During training, a neuron is *dropped out* with a probability p (e.g. p=0.5)



Deep Learning – Regularization - Dropout

Why is Dropout effective in generalizing better?

- Neurons cannot co-adapt
- They need to take into account more inputs



Convolutional Neural Networks

Mostly used for image processing: classification, localization, detection, segmentation

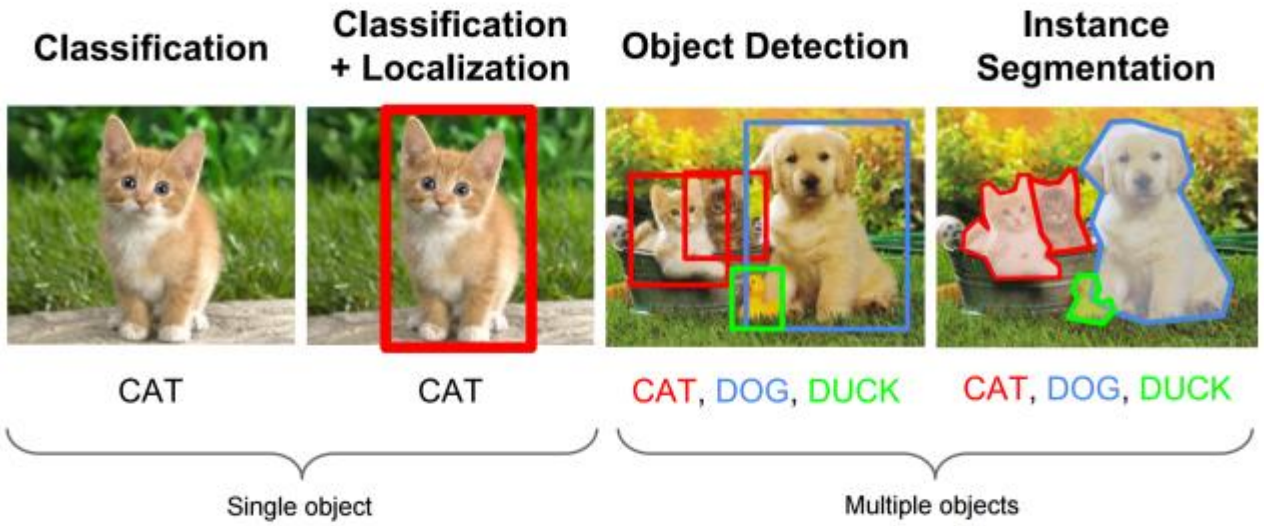


Image credit: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/object_localization_and_detection.html

Convolutional Neural Networks

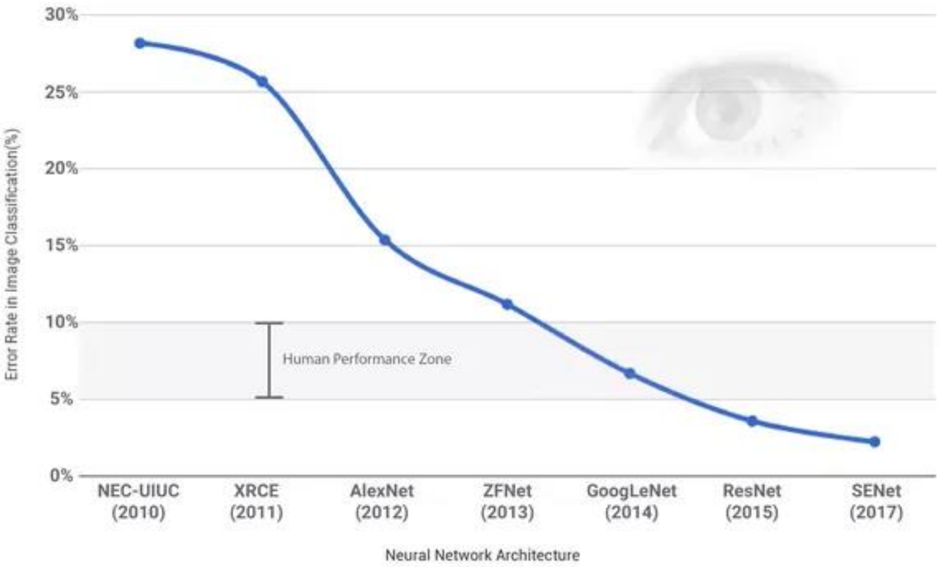


Image Classification

Convolutional Neural Networks

Also some funny applications: style transfer

+

=

Rain Princess, Leonid Afremov

Try by yourself on https://reiinakano.github.io/fast-style-transfer-deeplearnjs/

Convolutional Neural Networks

Face recognition

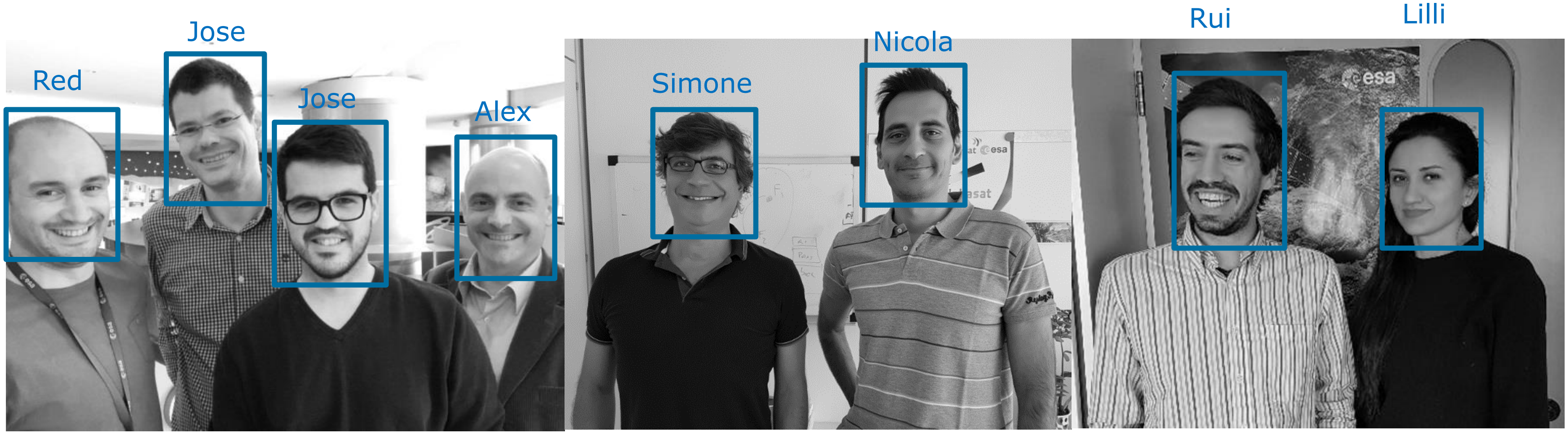
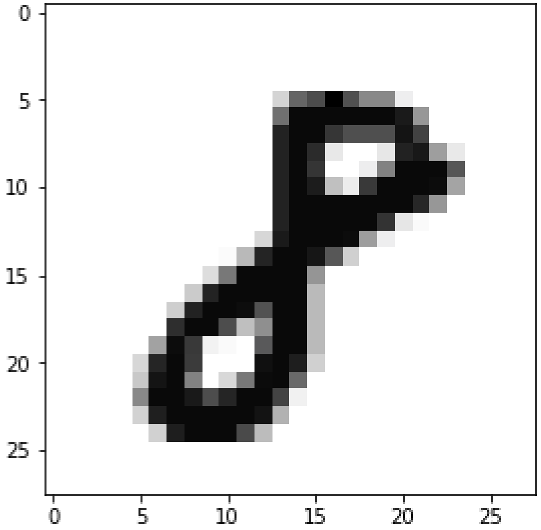


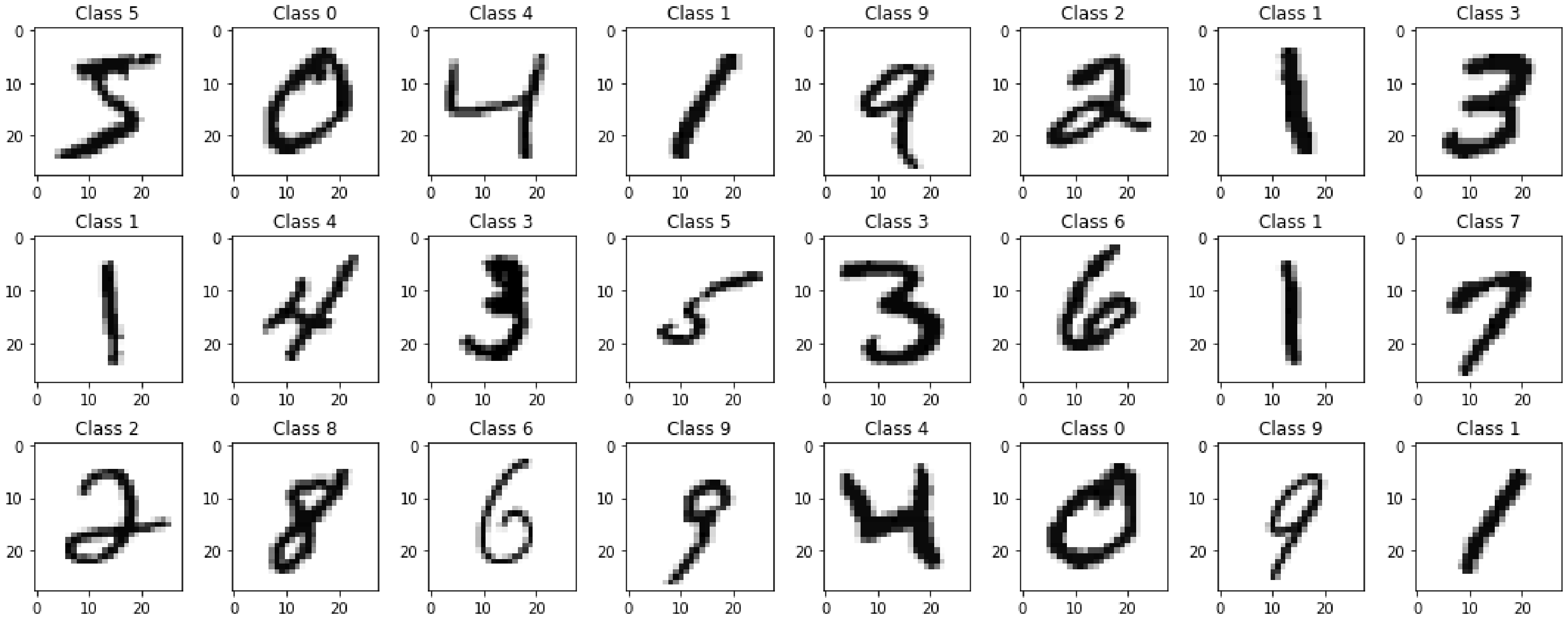
Image representation



0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	42	154	180	255	176	118	118	16	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	142	253	253	253	253	253	253	236	103	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	227	253	253	204	177	177	177	243	191	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	227	253	216	22	0	0	23	227	238	96	21	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	227	253	205	0	0	17	124	253	253	253	170	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	227	253	234	62	18	201	253	253	253	251	90	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	227	253	253	253	253	253	253	253	221	103	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	227	253	253	253	253	253	208	24	5	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	39	236	253	253	253	251	97	16	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	4	69	224	253	253	240	169	46	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	33	134	253	253	253	253	105	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	51	225	253	253	253	253	253	68	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	48	227	253	253	250	174	253	253	68	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	213	253	253	179	63	111	253	253	68	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	92	251	201	13	5	0	166	253	253	68	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	39	222	253	198	0	0	0	248	253	231	46	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	54	243	253	124	0	38	133	252	253	150	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	118	253	253	237	179	223	253	253	190	14	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	44	230	253	253	253	253	253	244	76	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	45	231	253	253	253	182	66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

28x28 pixel images corresponding to digits from the MNIST dataset https://en.wikipedia.org/wiki/MNIST_database

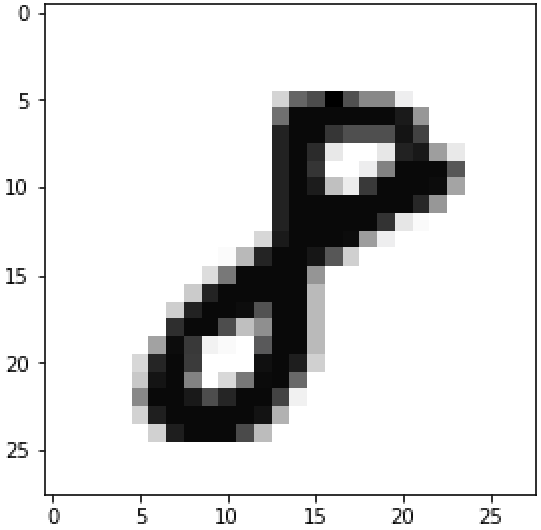
Some MNIST 28x28 pixel images

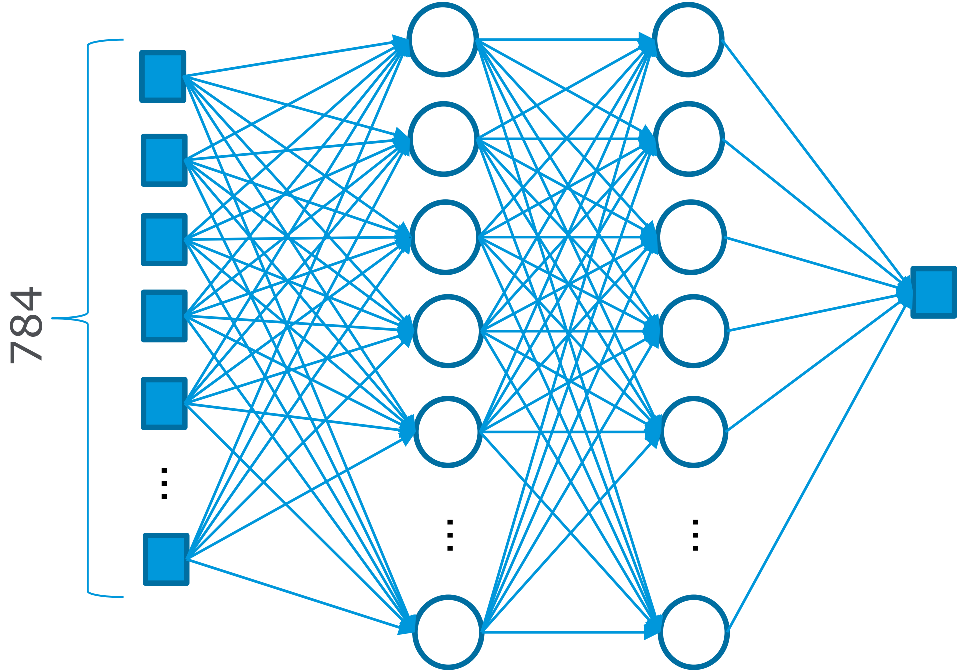


28x28 pixel images corresponding to digits from the MNIST dataset https://en.wikipedia.org/wiki/MNIST_database

(Deep) Neural Networks on images

Let's built a (Deep) Neural Network to tell 8 vs non-8

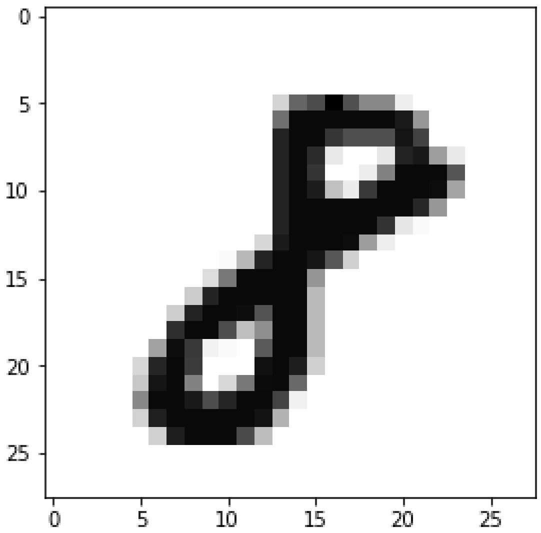




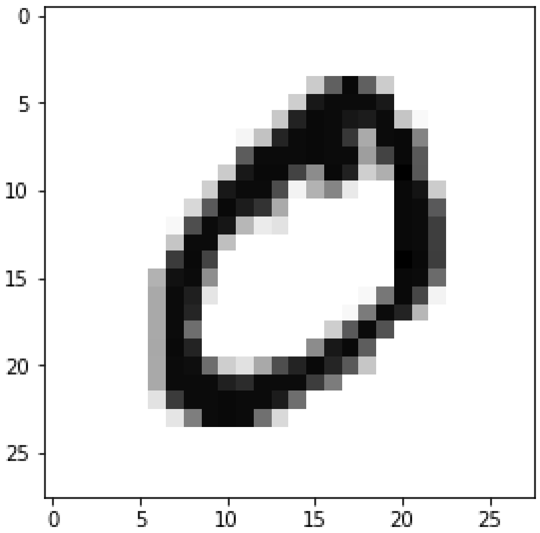
Input: 28x28=784 pixels

(Deep) Neural Networks on images

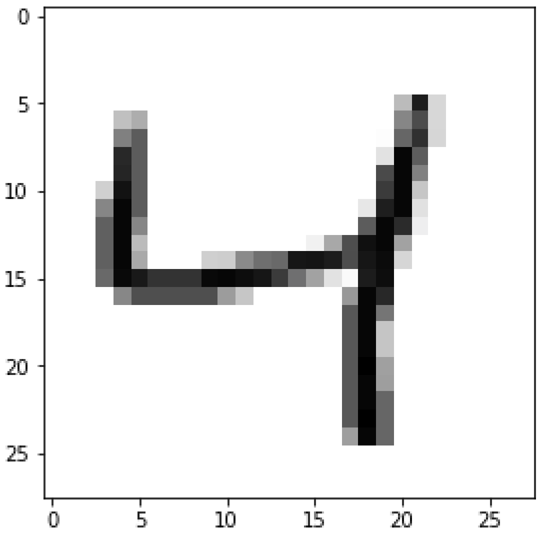
Despite its complexity, it works !



Output: 8



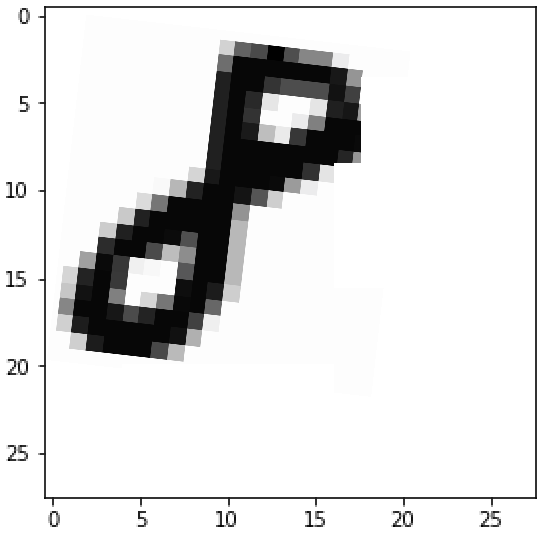
Output: non-8



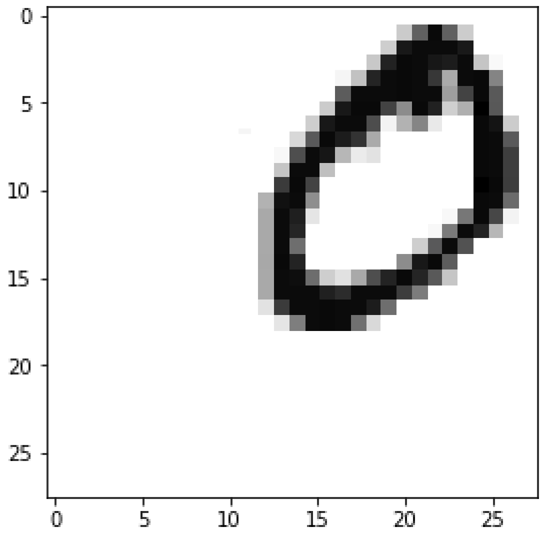
Output: non-8

(Deep) Neural Networks on images

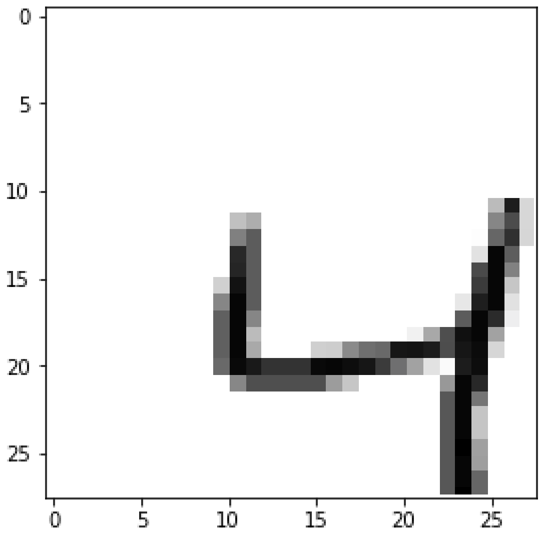
However, it depends very much on the exact location of the image



Output: ?



Output: ?



Output: ?

Convolution operation

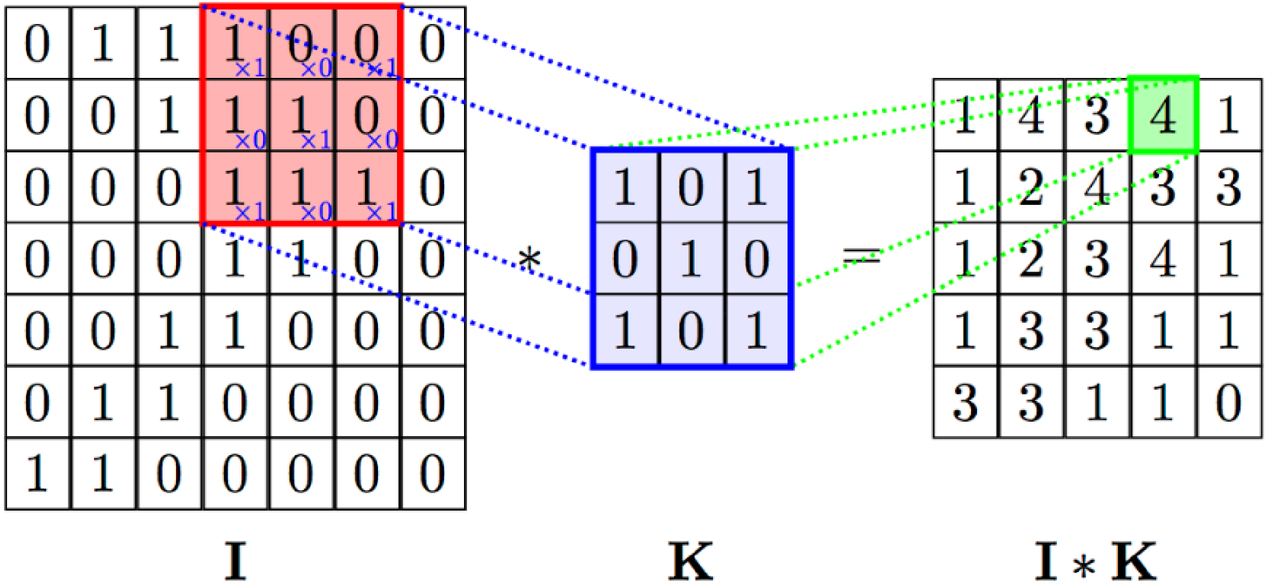


Image Kernel / Filter Convolution

Image credit: https://github.com/PetarV-/TikZ/tree/master/2D%20Convolution

Convolution intuition

Blur

1	1	1
1	1	1
1	1	1

Convolution intuition

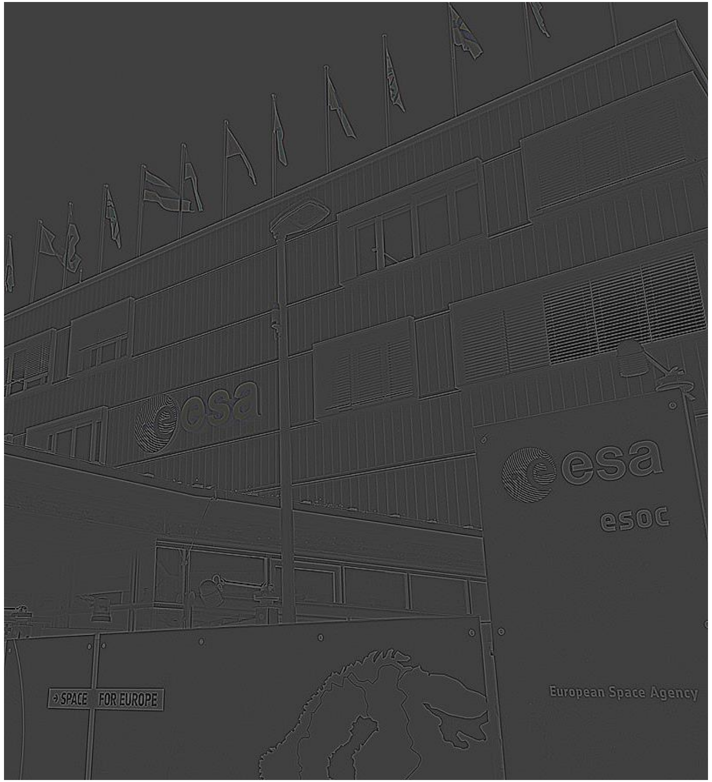
Sharpen

0	-1	0
-1	5	-1
0	-1	0

Convolution intuition

Detect Edges

0	1	0
1	-4	1
0	1	0



Convolutional Neural Networks

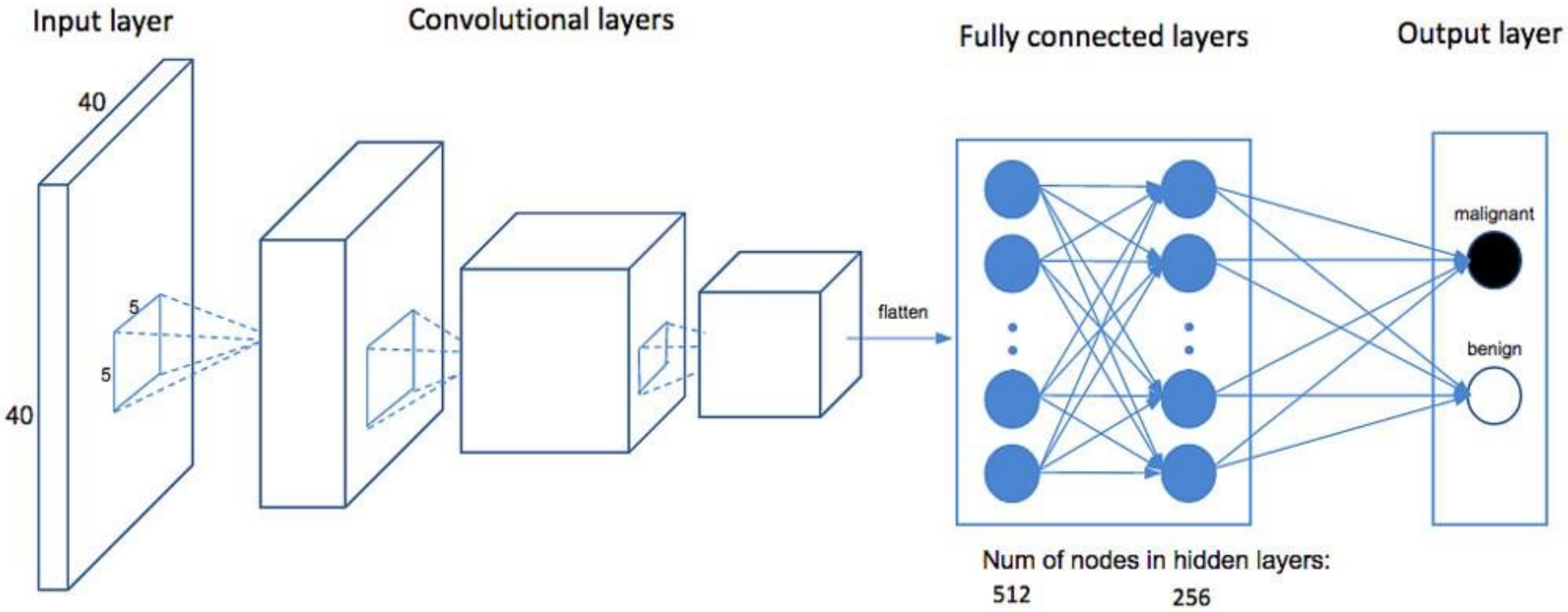


Image credit: https://blog.insightdatascience.com/automating-breast-cancer-detection-with-deep-learning-d8b49da17950

Convolutional Layer

- Need to specify how many kernels / filters of which size. Example: 32 filters of size (5x5)

- Usually followed by a Pooling layer

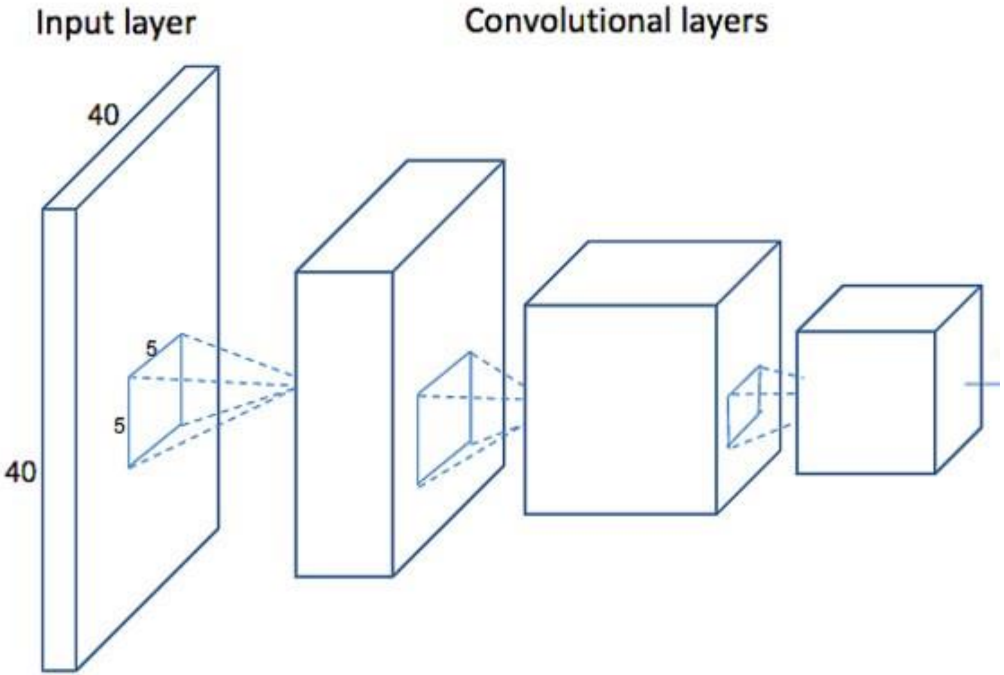


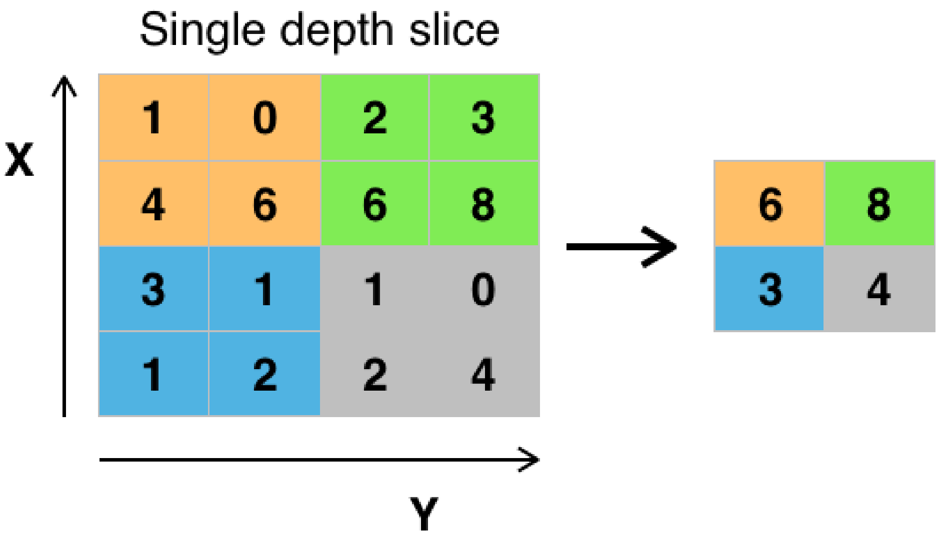
Image credit: https://blog.insightdatascience.com/automating-breast-cancer-detection-with-deep-learning-d8b49da17950

Pooling Layer

Subsample input to

- Reduce computation
- Reduce memory usage
- Reduce number of parameters

Max Pooling



Convolutional Neural Networks

In the Convolutional layers

- The parameters are the filters (kernels)
- The number of parameters is **independent from the image size**

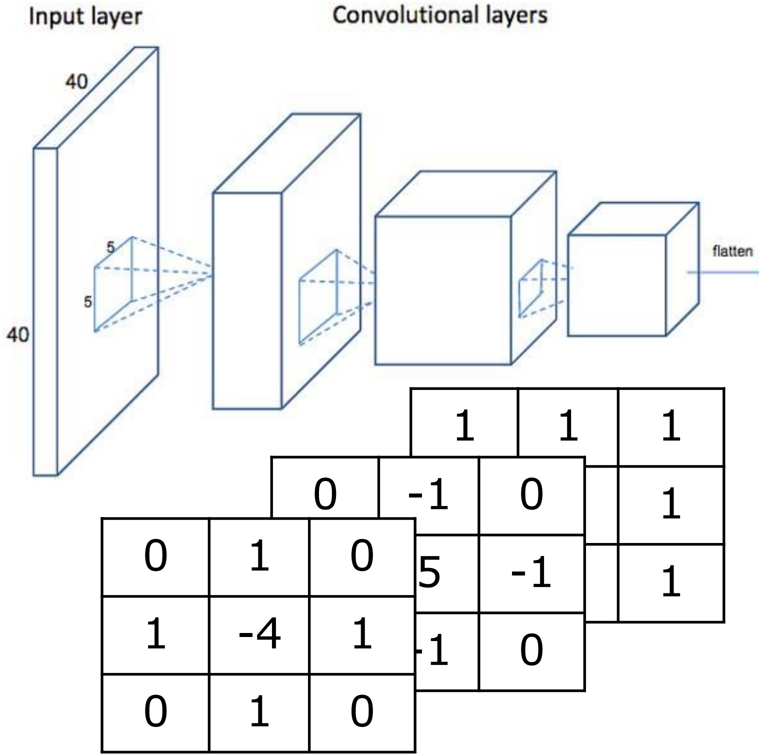


Image credit: https://blog.insightdatascience.com/automating-breast-cancer-detection-with-deep-learning-d8b49da17950

Convolutional Network – VGG16

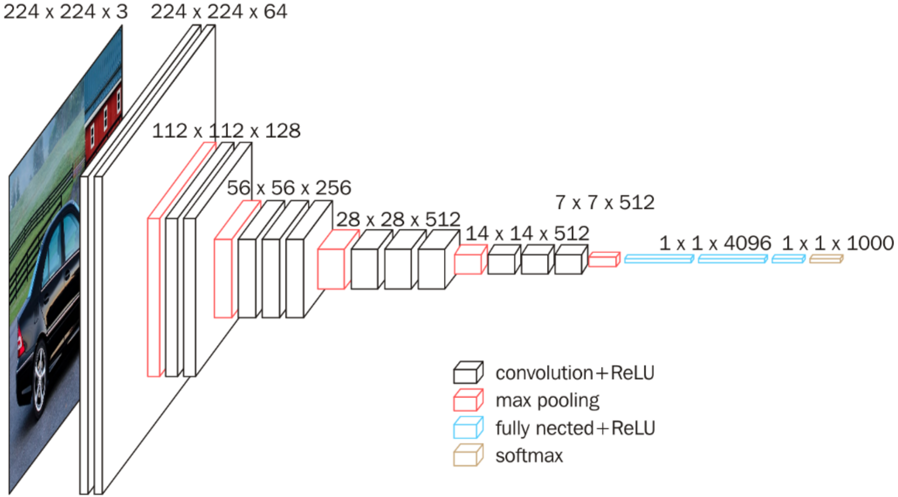


Image credit: https://www.packtpub.com/mapt/book/big_data_and_business_intelligence/9781786462961/9/ch09lvl1sec57/loading-a-pre-trained-model-to-speed-up-the-training

VGG16 – pre-trained model available

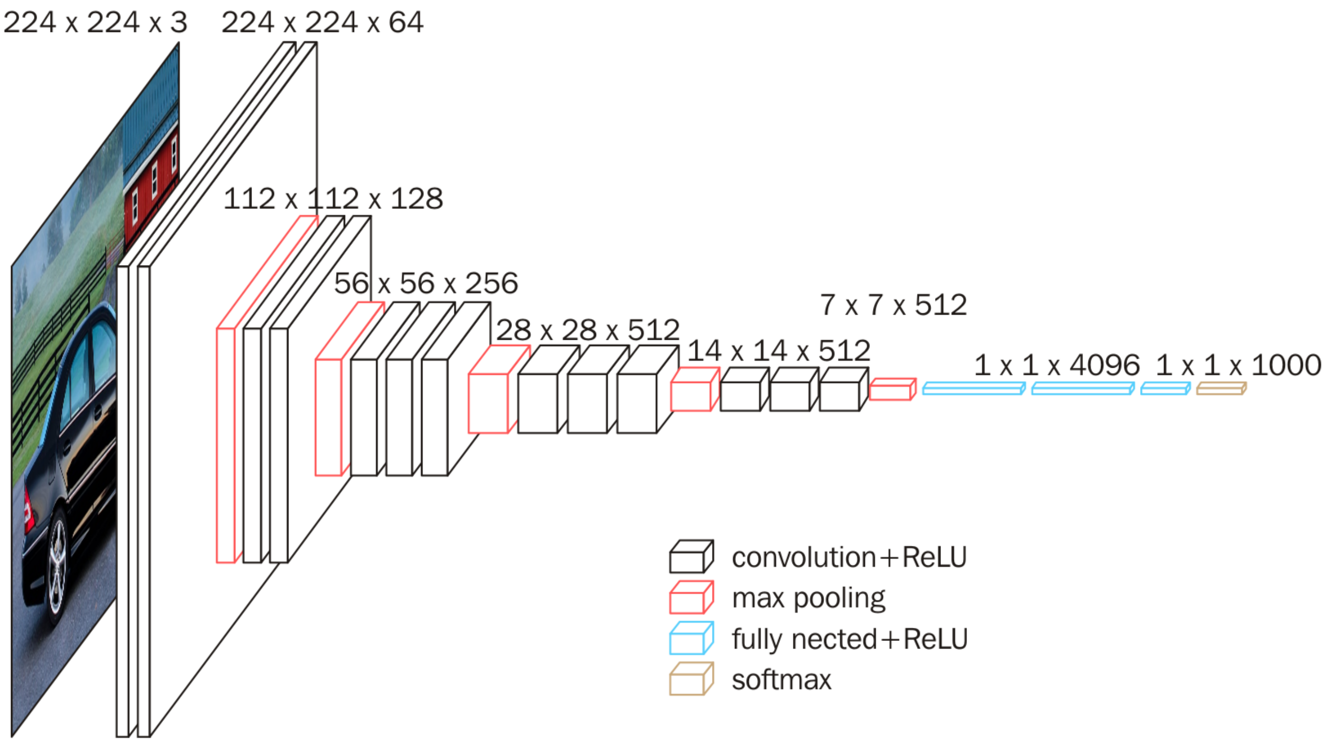


Image credit: https://www.packtpub.com/mapt/book/big_data_and_business_intelligence/9781786462961/9/ch09lvl1sec57/loading-a-pre-trained-model-to-speed-up-the-training

Transfer Learning using a pre-trained model

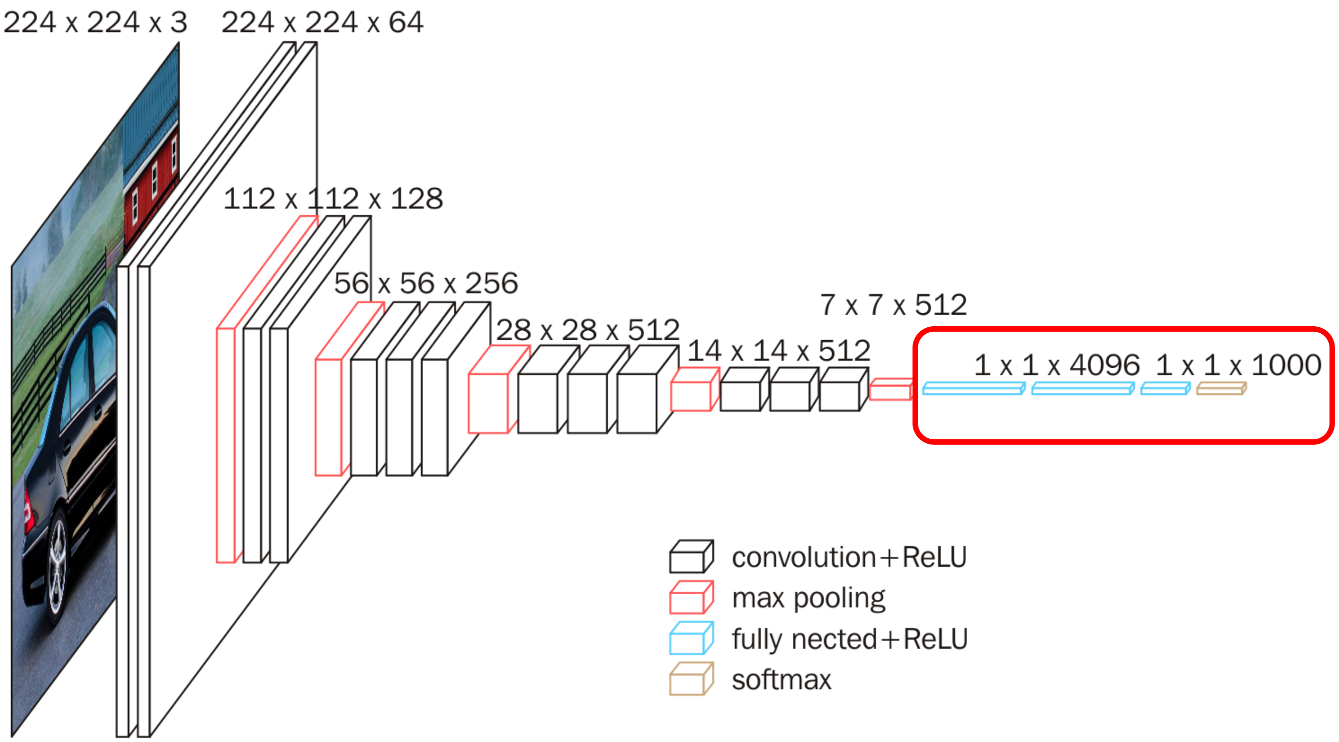
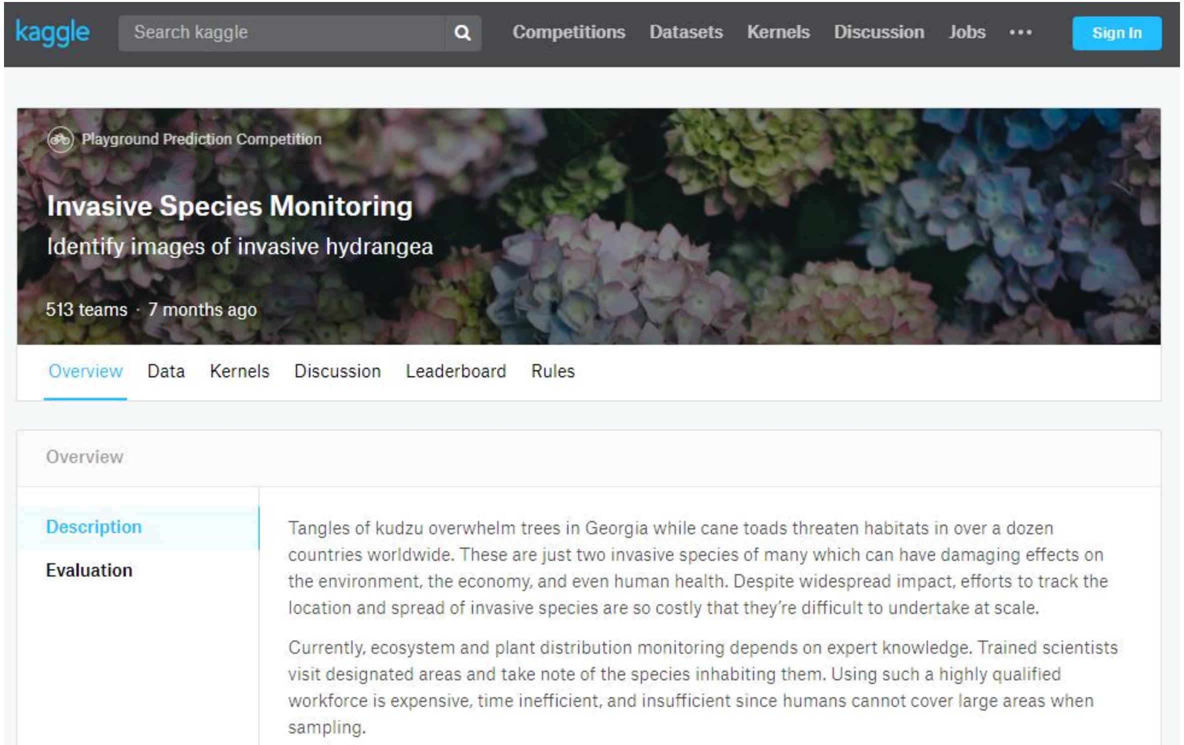


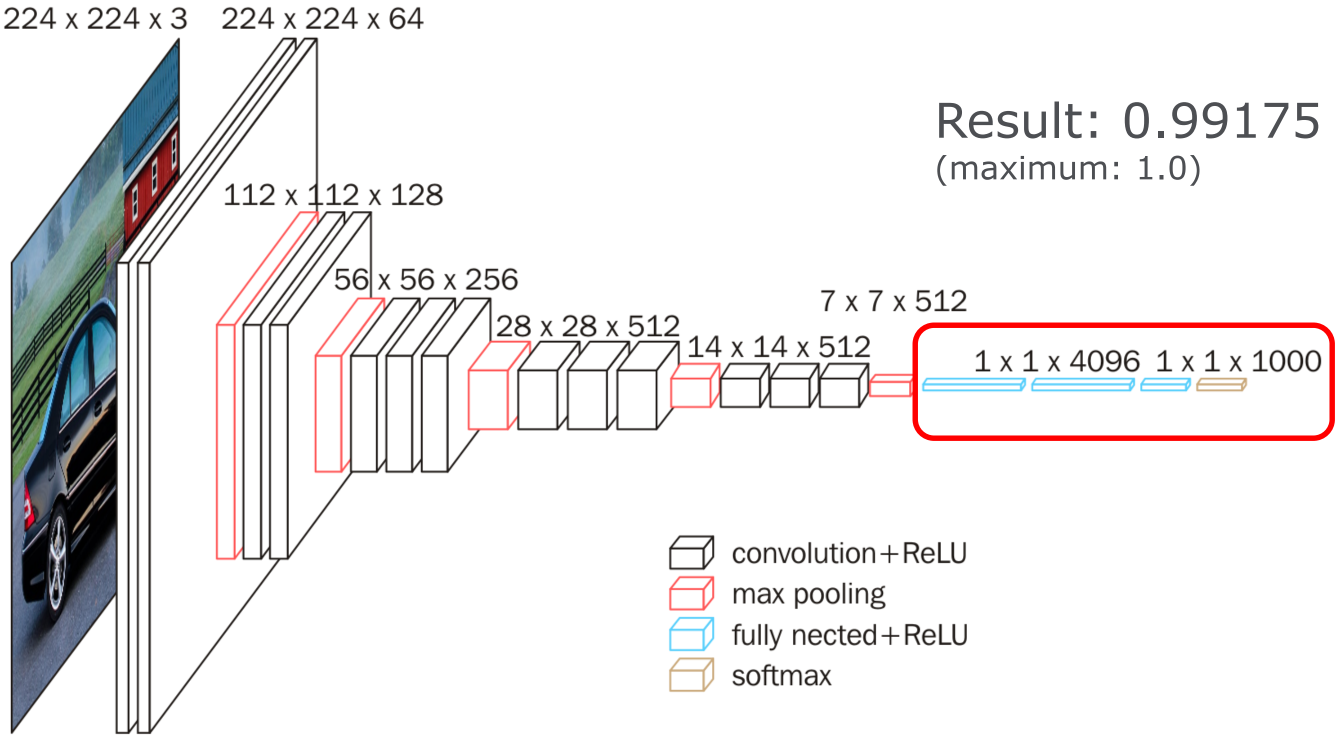
Image credit: https://www.packtpub.com/mapt/book/big_data_and_business_intelligence/9781786462961/9/ch09lvl1sec57/loading-a-pre-trained-model-to-speed-up-the-training

Transfer Learning – Kaggle example



https://www.kaggle.com/c/invasive-species-monitoring

Transfer Learning - Kaggle example



Result: 0.99175
(maximum: 1.0)

Image credit: https://www.packtpub.com/mapt/book/big_data_and_business_intelligence/9781786462961/9/ch09lvl1sec57/loading-a-pre-trained-model-to-speed-up-the-training

Transfer Learning - Kaggle example

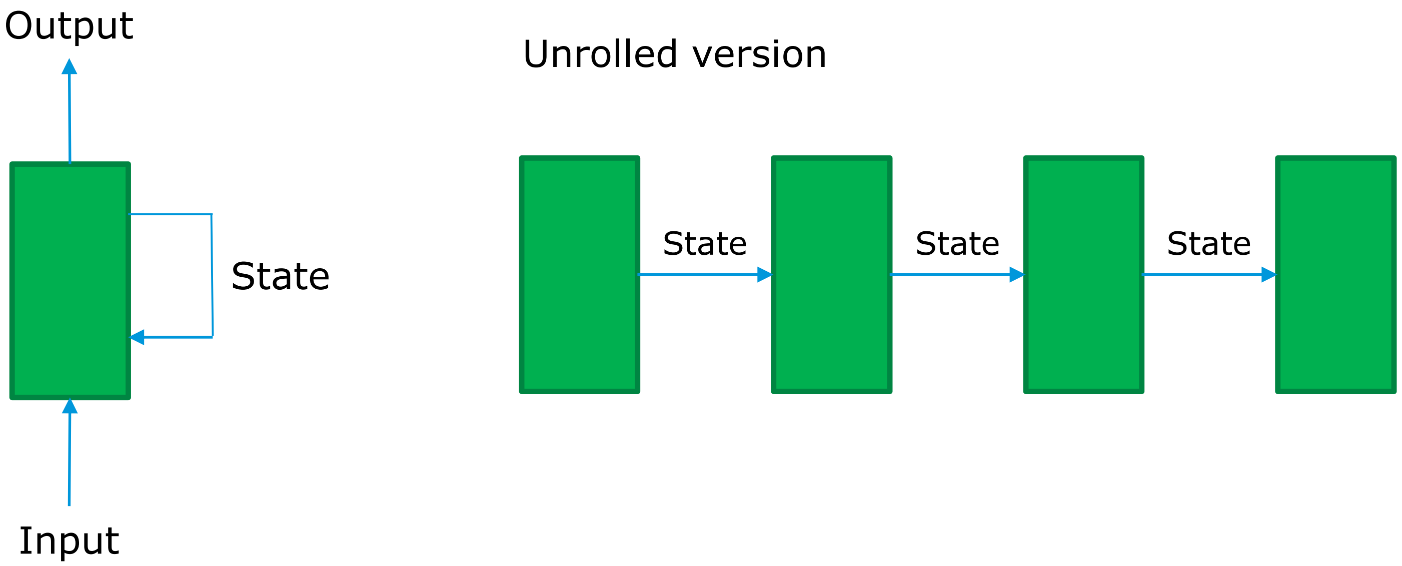
#	△1w	Team Name	Kernel	Team Members	Score	Entries	Last
1	—	James Requa			0.99770	56	7mo
2	—	Lefteris Fanioudakis			0.99767	140	7mo
3	—	Alexandre Cadrin-Chênevert			0.99643	30	7mo
4	—	Matteo Dunnhofer			0.99579	18	8mo
5	—	AndrewWang			0.99568	40	8mo
59	—	chetan reddy			0.99175	1	7mo
60	—	Jose Martinez			0.99175	20	10mo
61	—	Vincent Sim			0.99171	1	8mo
511	—						
512	—						
513	—						

https://www.kaggle.com/c/invasive-species-monitoring/leaderboard

Recurrent Neural Networks

Neural Networks used to model sequences

They have **memory** and are able to *remember* and *forget* as needed



Recurrent Neural Networks

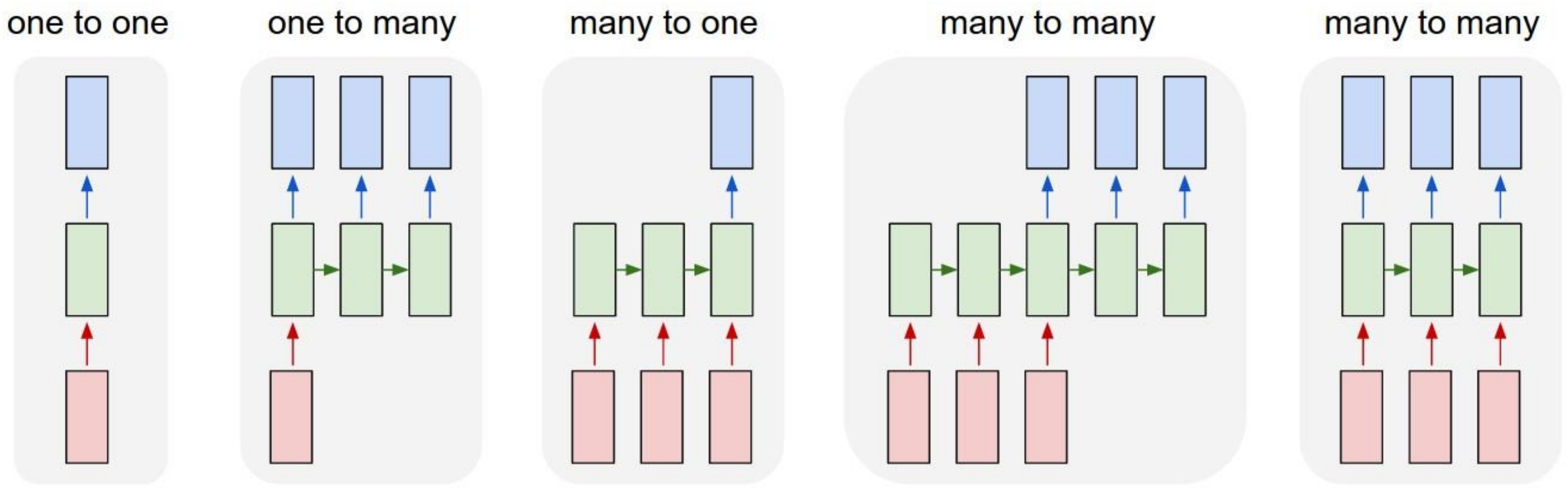
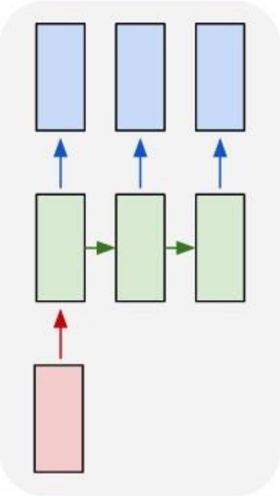


Image credit: https://karpathy.github.io/2015/05/21/rnn-effectiveness/

Recurrent Neural Networks – one to many

one to many



Example: Image Captioning

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

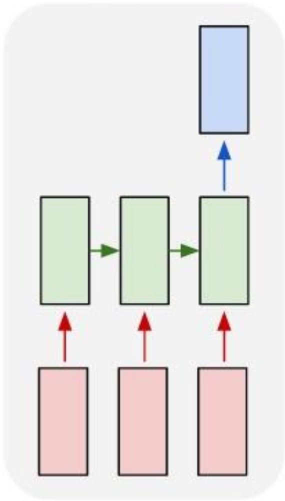
"two young girls are playing with lego toy."

Image credit: https://karpathy.github.io/2015/05/21/rnn-effectiveness/ https://cs.stanford.edu/people/karpathy/deepimagesent/

Recurrent Neural Networks

Example: predict rating from review text

many to one



★★★★★ One of the best Deep Learning books
By Sangyup Lee on February 26, 2018
Format: Paperback | **Verified Purchase**

One of the best Deep Learning books! For beginners, the theoretical explanations might seem somewhat insufficient, but, the practical parts are the best. You can be able to build your own networks easily after having read this book.

Comment | Was this review helpful to you? Yes No Report abuse

★★★★★ It is a very useful book, presenting the deep learning techniques on various ...
By Andrzej Stanislaw Kucik on March 16, 2018
Format: Paperback | **Verified Purchase**

It is a very useful book, presenting the deep learning techniques on various real-life examples, outlining exactly how to write a Python code to tackle them.

Comment | Was this review helpful to you? Yes No Report abuse

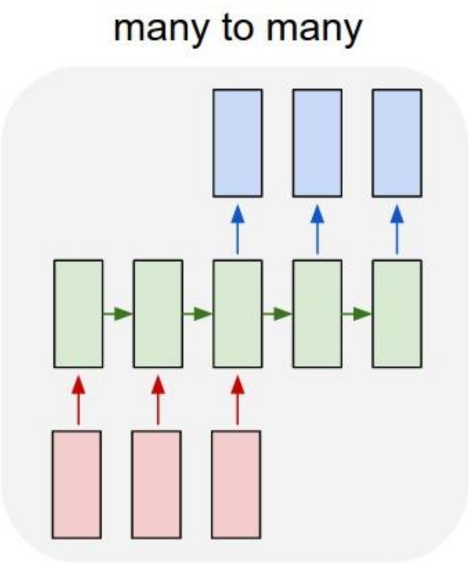
★★★★★ amazing book
By Ruru on March 6, 2018
Format: Paperback | **Verified Purchase**

amazing! a detailed guide on how to start deep learning projects using keras. i wish I had started deep learning with this book,

Comment | Was this review helpful to you? Yes No Report abuse

Image credit: https://karpathy.github.io/2015/05/21/rnn-effectiveness/
https://www.amazon.com/product-reviews/1617294438/ref=cm_cr_dp_d_cmps_btm?ie=UTF8&reviewerType=all_reviews

Recurrent Neural Networks



Example: language translation

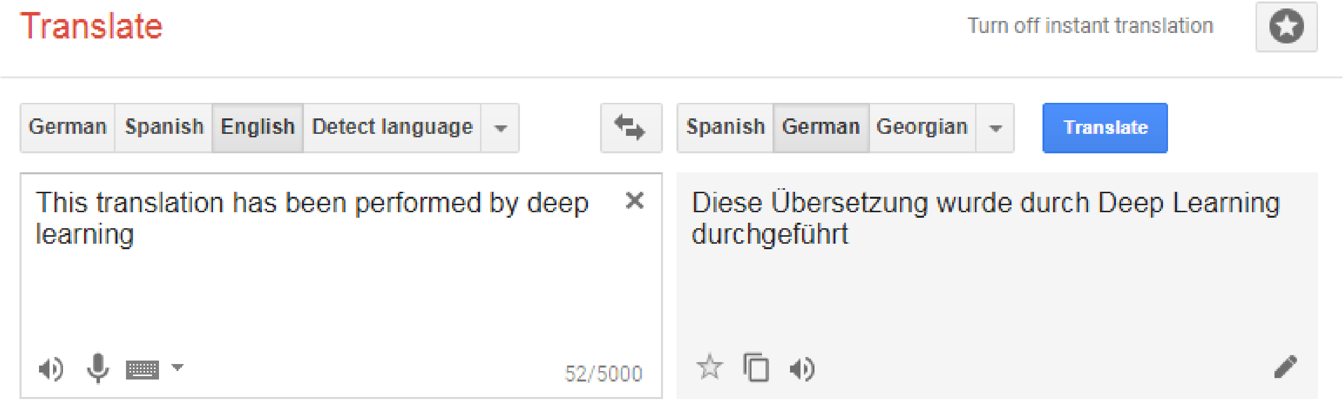
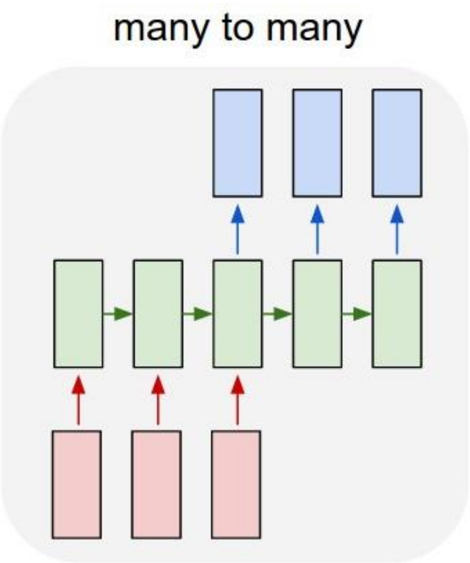


Image credit: https://karpathy.github.io/2015/05/21/rnn-effectiveness/

Recurrent Neural Networks



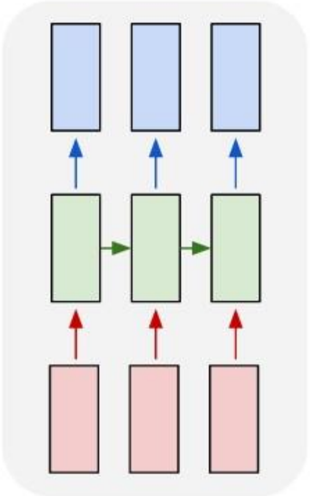
Example: chat bots

Machine: *could you please let me know what are the operating systems you are using ?*
Human: *i am using linux*
Machine: *ok it was in the ssl vpn on which you stated it was broken*
Machine: *please check if you are logged in to corp network*
Machine: *what is the error that you are running please*
Human: *i am seeing an error related to vpn*
Machine: *what is the error message that you are getting when connecting to vpn using network connect ?*

Image credit: https://karpathy.github.io/2015/05/21/rnn-effectiveness/ https://arxiv.org/abs/1506.05869v3

Recurrent Neural Networks

many to many



BUSINESS INSIDER UK

TECH

Artificial intelligence is attempting to write the next 'Game of Thrones' book

Fiona MacDonald, ScienceAlert
Aug. 29, 2017, 11:00 PM 683

If you're hanging out feverishly to find out what happens after this weeks' drama-filled season seven finale of *Game of Thrones*, you're not alone.

With the TV show now firmly ahead of the books, fans might have to wait until 2019 to see how George R.R. Martin's epic saga wraps up.

So to give us some much-needed new material to over-analyze in the meantime, an algorithm has started to write the sixth book for

The AI predicted some fan theories would come true concerning Daenerys and Jon. HBO

Image credit: https://karpathy.github.io/2015/05/21/rnn-effectiveness/
http://uk.businessinsider.com/ai-just-wrote-the-next-book-of-game-of-thrones-for-us-2017-8?IR=T

Summary

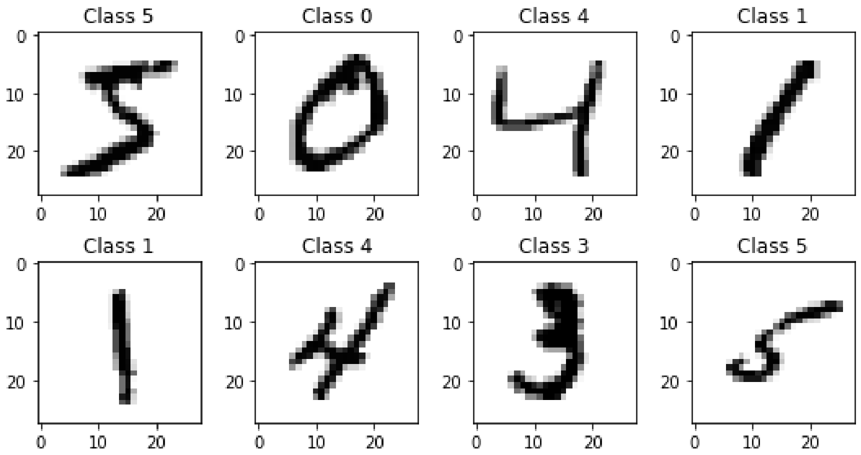
- Neural Networks
- Why Deep Neural Networks are better (usually)?
- How the vanishing gradient problem was solved
- Regularization (Dropout)
- Convolutional Neural Networks
- Recurrent Neural Networks

Materials: Slides, Code, Videos

Available on the Data Analytics ESA connect community

url: https://connect.esa.int/communities/community/data-analytics

Hands on: image classification



What is next?

April 11th 16:00 – Press Room (was H-I)

Session 5: Unsupervised Learning

- Clustering
- PCA
- Auto-encoders
- Semi-supervised learning: DrMUST, Novelty Detection

- Hands on

Resources

Watch the video of this lecture

https://dlmultimedia.esa.int/download/public/videos/2048/03/008/4803_008_AR_EN.mp4

Watch the practical exercise video

https://dlmultimedia.esa.int/download/public/videos/2048/03/007/4803_007_AR_EN.mp4

Get presentation and additional resources on

https://github.com/jmartinezheras/2018-MachineLearning-Lectures-ESA

Thank you

Data Analytics Team for Operations (DATO)

Jose Martinez Heras

LinkedIn: https://www.linkedin.com/in/josemartinezheras/