Artificial neural networks

Mads Jensen, PhD

mads@cas.au.dk

November 16th, 2020

AARHUS UNIVERSITY

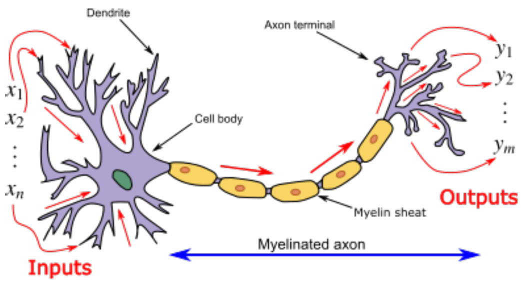
IMC INTERACTING MINDS CENTRE

CFIN

Contents

Artificial neural networks

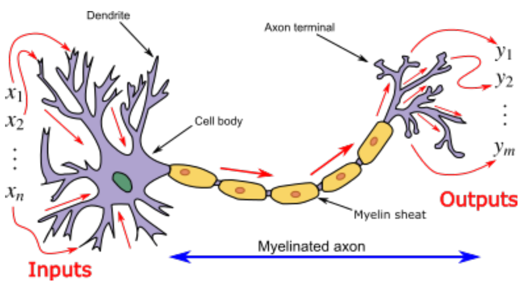
(Artificial) neurons



Real neuron:

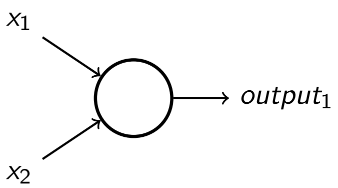
- connected to other neurons
- can perform simple computations
- inhibition and exhibition

(Artificial) neurons



Real neuron:

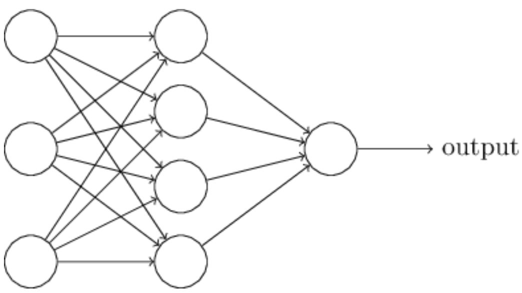
- connected to other neurons
- can perform simple computations
- inhibition and exhibition



Artificial neuron:

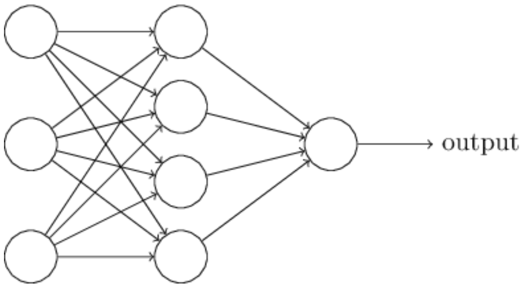
- connected to other neurons
- can perform simple computations
- inhibition and exhibition

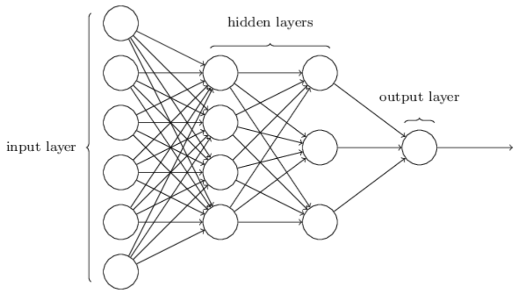
Neural network



(Figure from Nielsen, 2015)

Neural network





(Figure from Nielsen, 2015)

XOR (exclusive or)

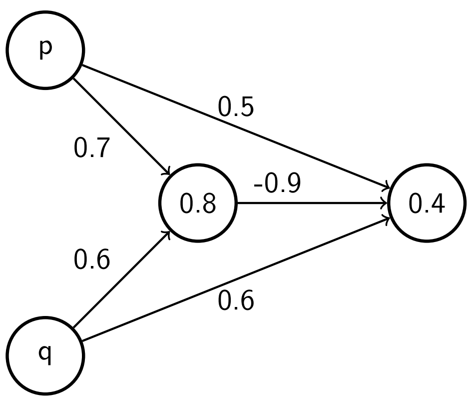
p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

$$p \oplus q = (p \vee q) \wedge \neg(p \wedge q)$$

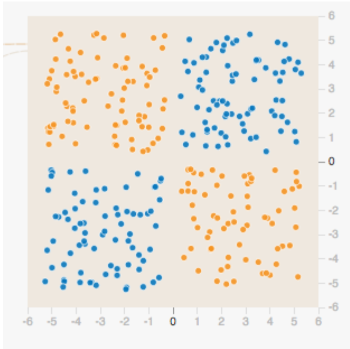
XOR (exclusive or)

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

$$p \oplus q = (p \vee q) \wedge \neg(p \wedge q)$$



XOR (exclusive or)



(Figure from https://playground.tensorflow.org/)

Machine learning

	Supervised learning
Data	$\{X, y\}$
Model	$y \approx f_\theta(X)$
Loss/cost	$\mathcal{L}(\theta) = \sum_{i=1}^{N} l(f_\theta(x_i), y_i)$
Optimisation	$\theta^* = \arg\min_\theta \mathcal{L}(\theta)$

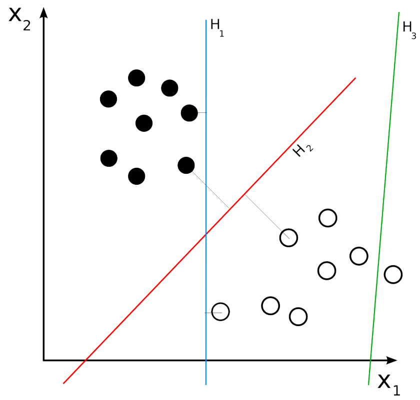
Neural network parts

- Linear layer
- Activation function
- Loss/cost function

- Neurons: units
- Parameters: weights

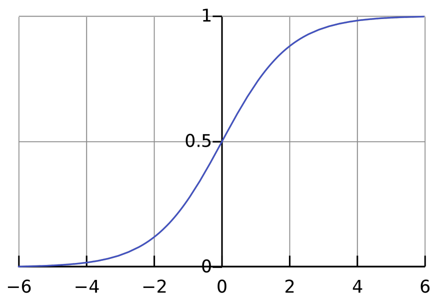
Linear layer

$f_{\text{linear}}(x, W, b) = Wx + b$

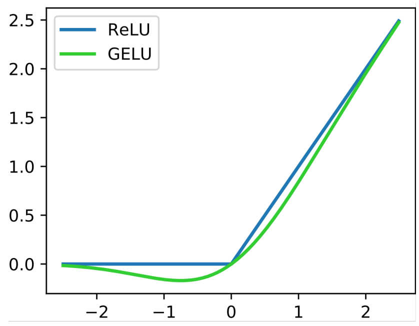


(Figure from https://en.wikipedia.org/wiki/Linear_classifier)

Activation functions



Sigmoid function
$S(x) = \frac{1}{1+e^{-x}}$



Rectified **L**inear **U**nit (ReLU)
$f(x) = max(0, x)$

(Figure from https://en.wikipedia.org/wiki/Rectifier_(neural_networks))

Softmax

Softmax is a generalization of the logistic function to multiple dimensions.

$$f_{sm}(x) = \frac{e^{x}}{\sum_{j=1}^{k} e^{x_j}}$$

Often used as the last layer before the loss/cost function

Loss/cost function

Cross entropy (also called logarithmic loss, log loss, or logistic loss)

$$H(p, q) = -\sum_i p_i \log_2(q_i)$$

Loss/cost function

Cross entropy (also called logarithmic loss, log loss, or logistic loss)

$$H(p, q) = -\sum_i p_i \log_2(q_i)$$

Shannon entropy (from lecture 5):

$$\mathrm{H}(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i)$$

Learning

- No learning yet!

Learning

- No learning yet!
- Building block for learning
  - Function ($y \approx f_\theta(X)$)
  - Loss/cost
  - Optimisation
  - Data

Learning

- No learning yet!
- Building block for learning
  - Function ($y \approx f_\theta(X)$)
  - Loss/cost
  - Optimisation
  - Data

What is learning in this context?

Learning

- No learning yet!
- Building block for learning
  - Function ($y \approx f_\theta(X)$)
  - Loss/cost
  - Optimisation
  - Data

What is learning in this context?

- Changing response/output to a fixed stimulus

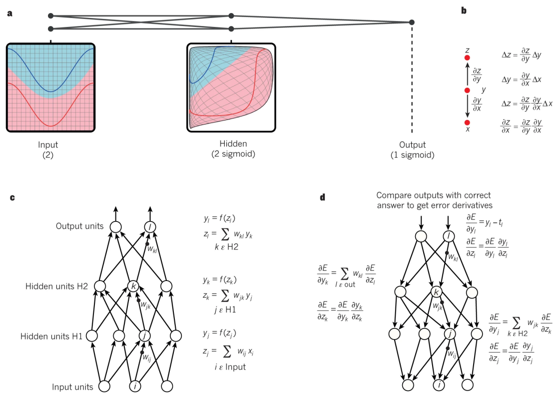
Learning

- No learning yet!
- Building block for learning
  - Function ($y \approx f_\theta(X)$)
  - Loss/cost
  - Optimisation
  - Data

What is learning in this context?

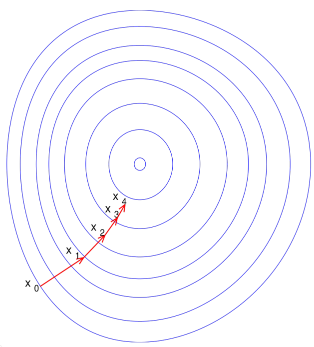
- Changing response/output to a fixed stimulus
- Response *after* seeing data is not the same as *before* seeing data.

Backpropagation



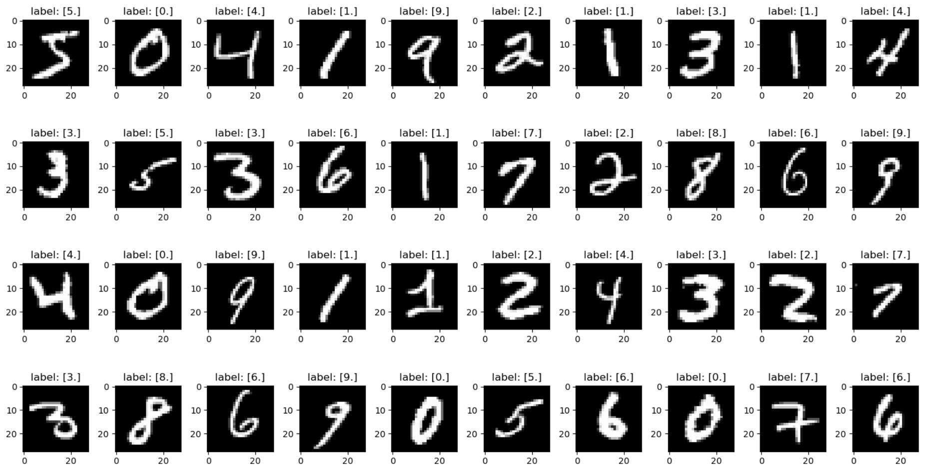
(Figure from LeCun et al., 2015)

Gradient decent



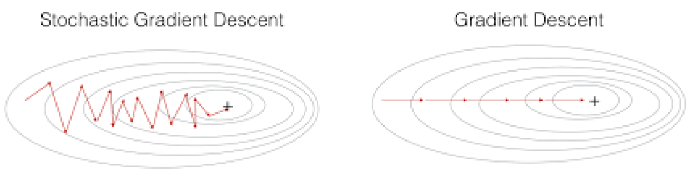
(Figure from https://en.wikipedia.org/wiki/Gradient_descent)

Mini batches



Data from http://yann.lecun.com/exdb/mnist/

Stochastic gradient decent



(Figure from https://medium.com/bayshore-intelligence-solutions/why-is-stochastic-gradient-descent-2c17baf016de)

Deep learning

Deep learning

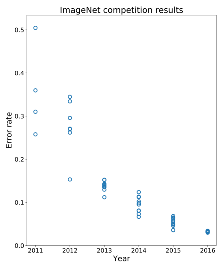
- Deep learning
- Deep neural networks

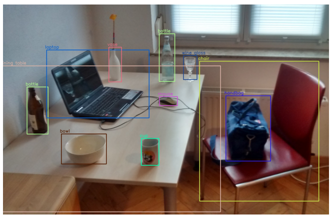
Deep learning

- Deep learning
- Deep neural networks
- Deep vs shallow neural networks
  - Shallow <= 3 hidden layers
  - Deep > 3 hidden layers

Deep learning

- Deep learning
- Deep neural networks
- Deep vs shallow neural networks
  - Shallow <= 3 hidden layers
  - Deep > 3 hidden layers





(Figures from https://en.wikipedia.org/wiki/Object_detection and https://en.wikipedia.org/wiki/ImageNet)

Technologies

Technologies

(Figure from https://upload.wikimedia.org/wikipedia/en/a/a7/Quake_gameplay.png)

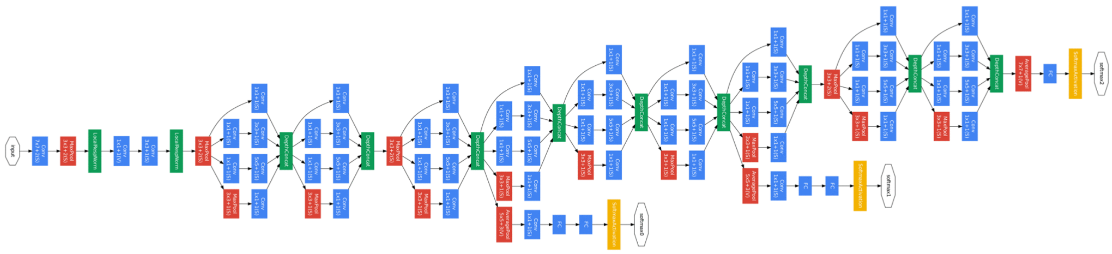
Technologies

- Graphical processing unit (GPU)

Technologies

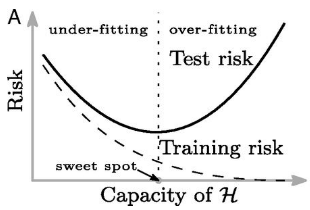
- Graphical processing unit (GPU)
- Data
  - Internet
  - Facebook, Twitter etc.
  - Deep learning need a lot of training data

GoogLeNet



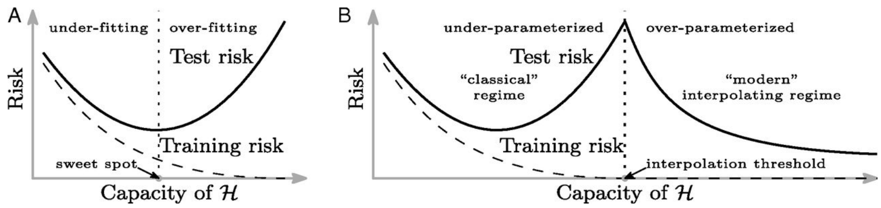
(Figure from Szegedy et al., 2015)

Underfitting & overfitting



(Figure from Belkin et al., 2019)

Underfitting & overfitting

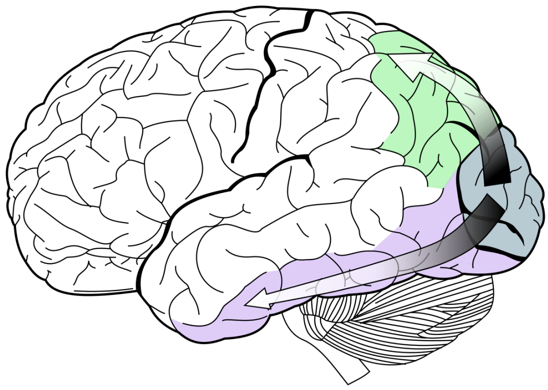


(Figure from Belkin et al., 2019)

Convolutional neural networks

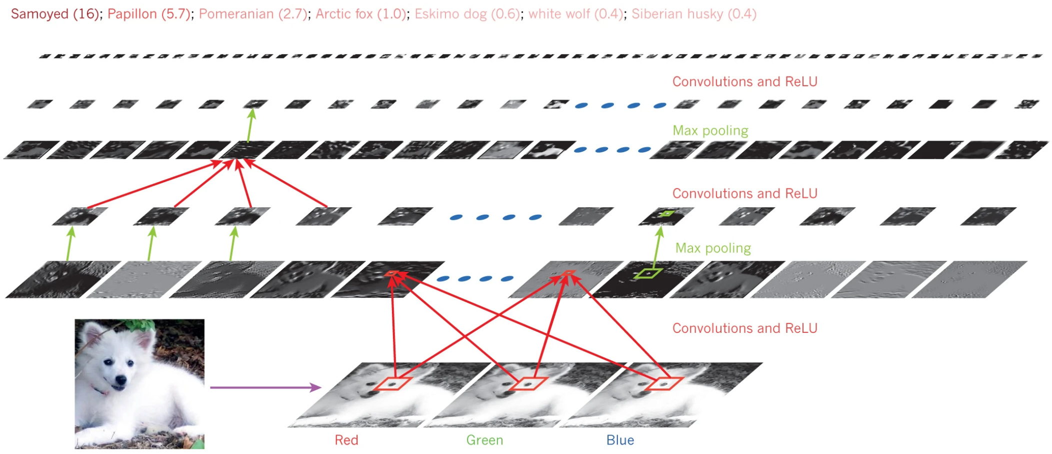
(Figure from https://en.wikipedia.org/wiki/Tree)

Convolutional neural networks



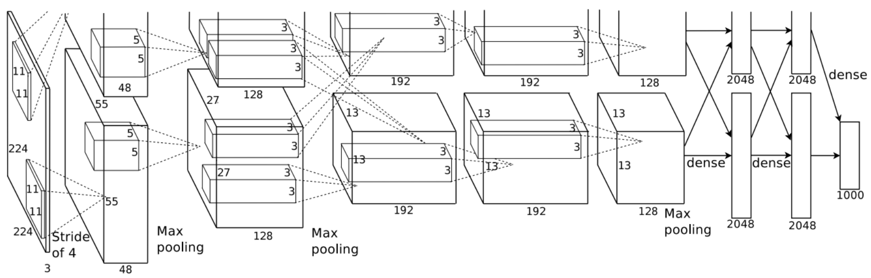
(Figure from https://en.wikipedia.org/wiki/Visual_perception)

Convolutional neural networks



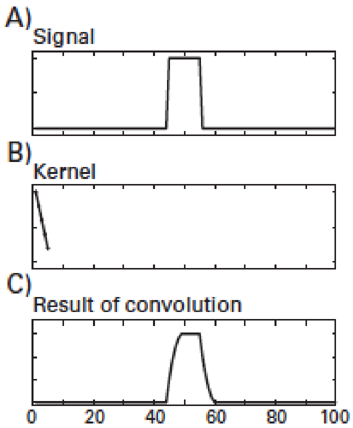
(Figure from LeCun et al., 2015)

AlexNet



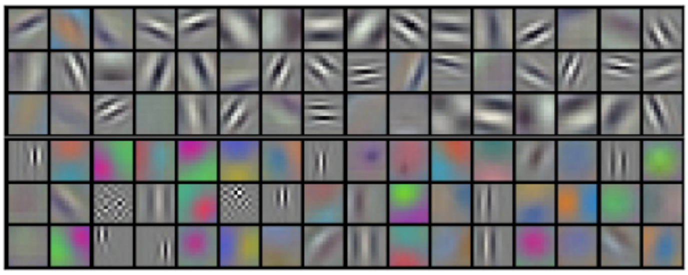
(Figure from Krizhevsky et al., 2017)

Convolution



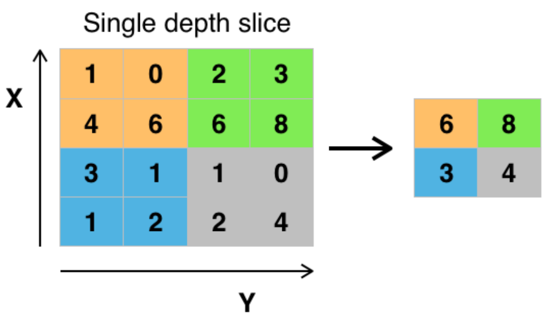
(Figure from Cohen, 2014)

Convolution kernels



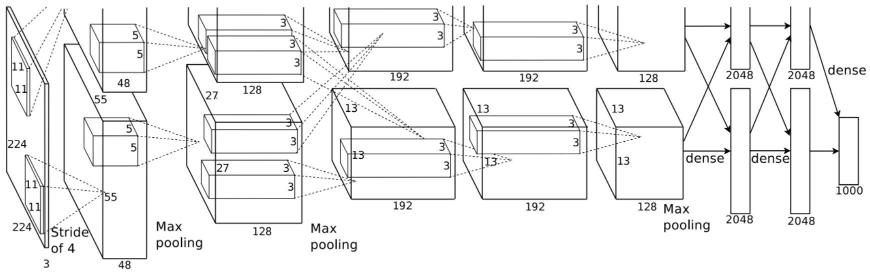
(Figure from Krizhevsky et al., 2017)

Max pooling



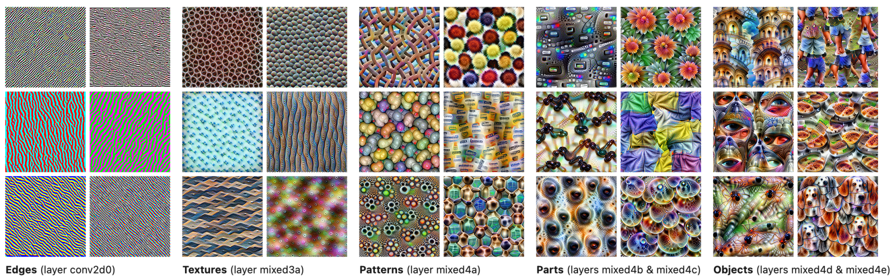
(Figure from https://en.wikipedia.org/wiki/Convolutional_neural_network)

AlexNet



(Figure from Krizhevsky et al., 2017)

Feature visualisation



(Figure from Olah et al., 2017)

Problems with max pooling

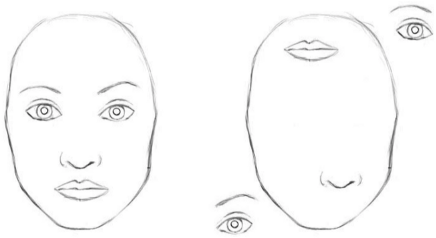
> *The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster.*
>
> *Hinton, reddit AMA*[1]

[1] https://www.reddit.com/r/MachineLearning/comments/2lmo0l/ama_geoffrey_hinton/

Problems with max pooling

The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster.

Hinton, reddit AMA[1]

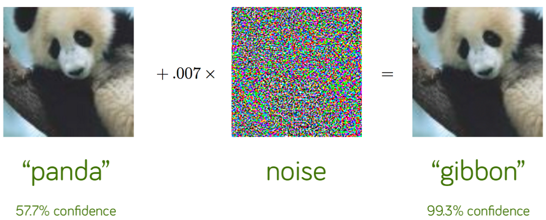


(Figure from https://medium.com/ai%C2%B3-theory-practice-business/understanding-hintons-capsule-networks-part-i-intuition-b4b559d1159b)

[1] https://www.reddit.com/r/MachineLearning/comments/2lmo0l/ama_geoffrey_hinton/

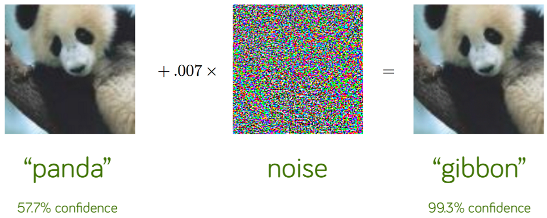
Adversarial attacks

Adversarial attacks



(Top figure from Goodfellow et al (2015);

Adversarial attacks



(Top figure from Goodfellow et al (2015); bottom from Eykholt et al. (2018))

"Assignment" for next lecture

What to do in last lecture?

In the syllabus for the last lecture:

- summary
- Q & A

Consider if there is anything you would like to have covered

Questions?

1. Artificial neural networks
   - Linear layer
   - Activation functions
   - Loss/cost function
   - Learning
2. Deep learning
3. Convolutional neural networks
   - Convolution
   - Max pooling
4. Adversarial attacks

References I

Belkin, M., Hsu, D., Ma, S., & Mandal, S. (2019). Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, *116*(32), 15849–15854. https://doi.org/10.1073/pnas.1903070116

Cohen, M. X. (2014). *Analyzing neural time series data: Theory and practice*. The MIT Press.

Eykholt, K., Evtimov, I., Fernandes, E., Li, B., Rahmati, A., Xiao, C., Prakash, A., Kohno, T., & Song, D. (2018). Robust Physical-World Attacks on Deep Learning Visual Classification. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1625–1634. https://doi.org/10.1109/CVPR.2018.00175

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2015, March 20). *Explaining and Harnessing Adversarial Examples*. arXiv: 1412.6572 [cs, stat]. Retrieved November 15, 2020, from http://arxiv.org/abs/1412.6572

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

References II

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444. https://doi.org/10.1038/nature14539

Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 2018). Determination press San Francisco, CA. http://neuralnetworksanddeeplearning.com/

Olah, C., Mordvintsev, A., & Schubert, L. (2017). Feature visualization. *Distill*. https://doi.org/10.23915/distill.00007

Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9. https://doi.org/10.1109/CVPR.2015.7298594