

Neural networks cont.

Mads Jensen, PhD

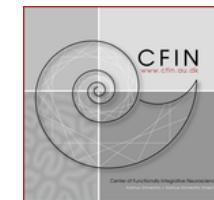
✉ Mads@cas.au.dk



AARHUS UNIVERSITY



 **IMC**
INTERACTING MINDS CENTRE



Contents

1. Recurrent neural networks
2. Reinforcement learning
 1. Pong
 2. AlphaGo
3. Networks as models
4. Generative Adversarial Networks (GANs)
5. Neural network for EEG
6. Last lecture

Neural networks recap

Supervised learning

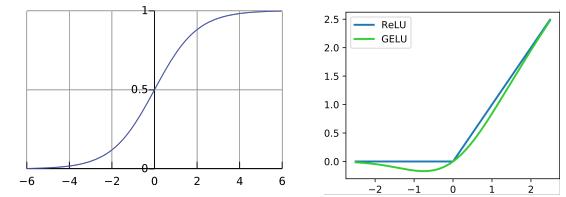
Data $\{X, y\}$

Model $y \approx f_\theta(X)$

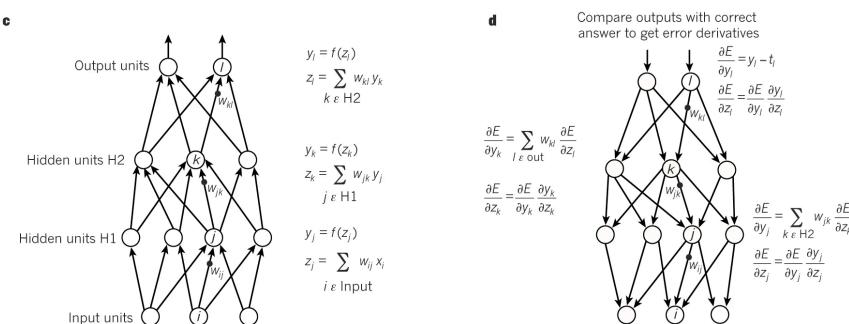
$$\text{Loss/cost} \quad \mathcal{L}(\theta) = \sum_{i=1}^N l(f_\theta(x_i), y_i)$$

$$\text{Optimisation} \quad \theta^* = \arg \min_{\theta} \mathcal{L}(\theta)$$

- Linear layer
- Activation function
- Loss/cost functions

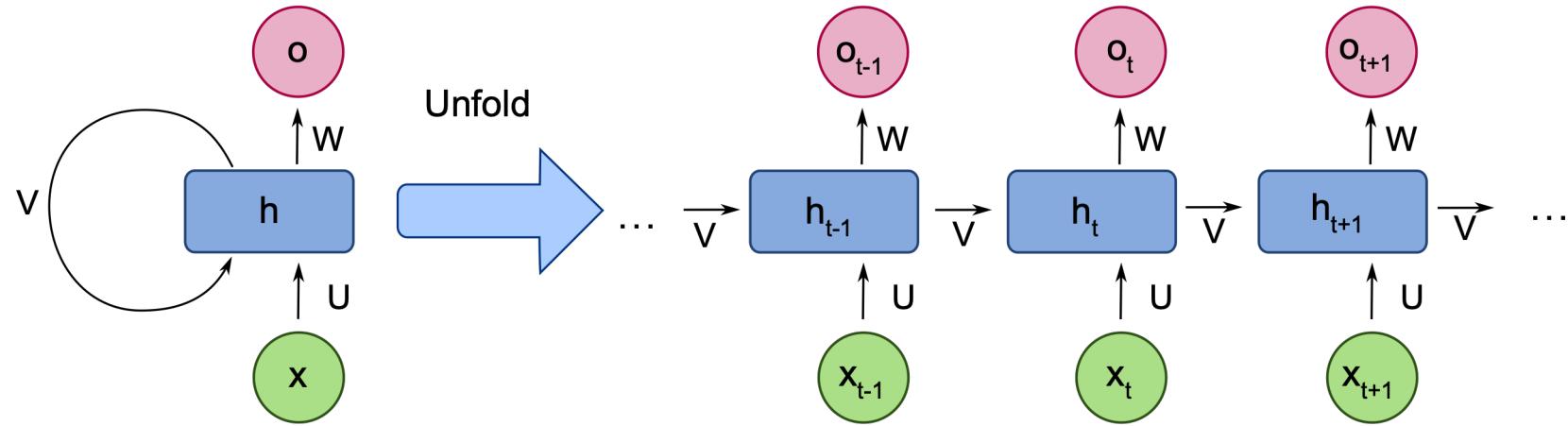


(Figure from
[https://en.wikipedia.org/wiki/Rectifier_\(neural networks\)](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)))



(Figure from LeCun et al., 2015)

Recurrent neural networks



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

Recurrent neural networks

Supervised learning

Data

$$\{X, y\}$$

Sequence modelling

Model

$$y \approx f_\theta(X)$$

$$p(X) \approx f_\theta(X)$$

Loss/cost

$$\mathcal{L}(\theta) = \sum_{i=1}^N l(f_\theta(x_i), y_1)$$

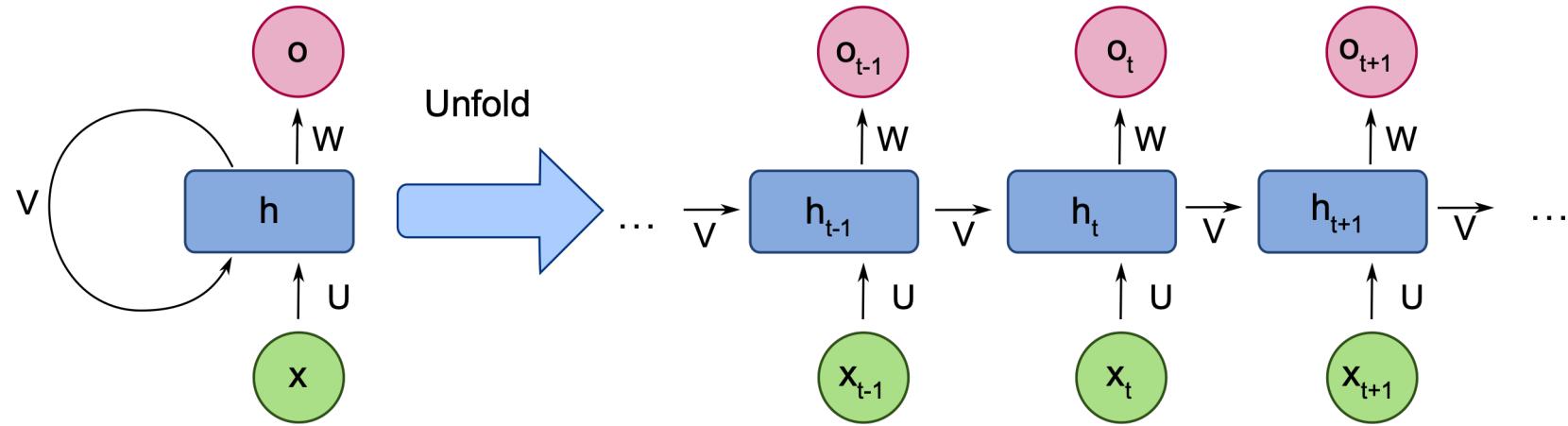
$$\mathcal{L}(\theta) = \sum_{i=1}^N \log p(f_\theta(x_i))$$

Optimisation

$$\theta^* = \arg \min_{\theta} \mathcal{L}(\theta)$$

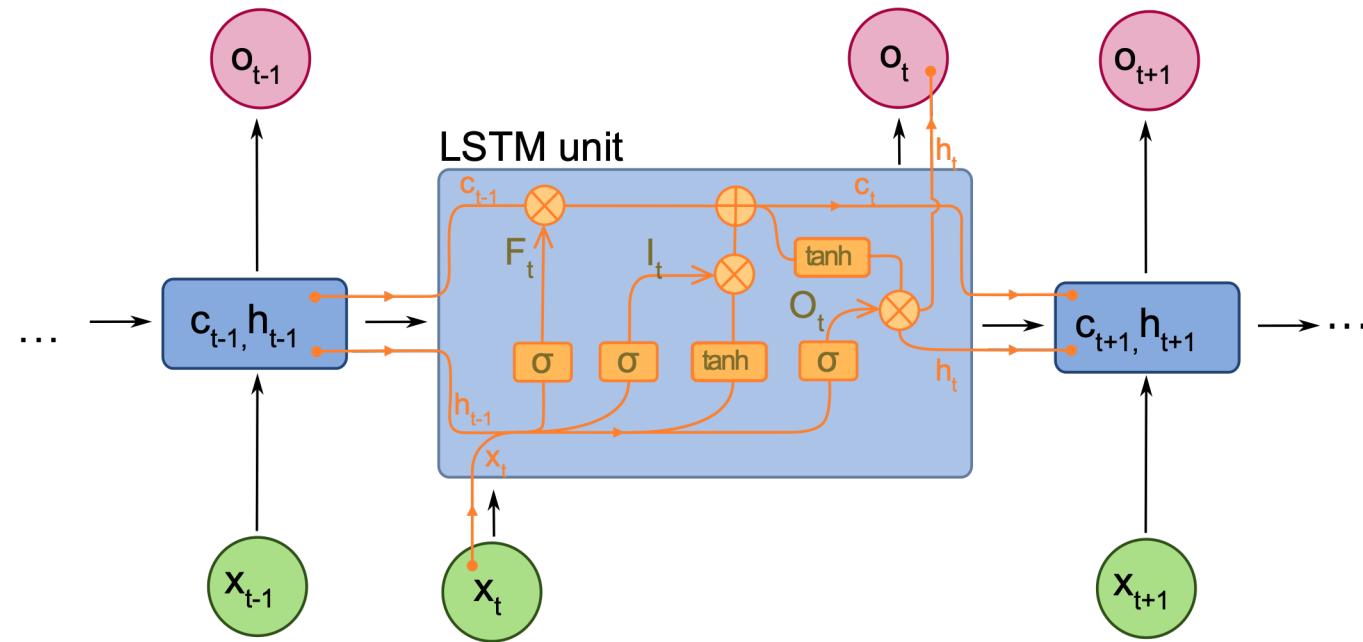
$$\theta^* = \arg \max_{\theta} \mathcal{L}(\theta)$$

Recurrent neural networks



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

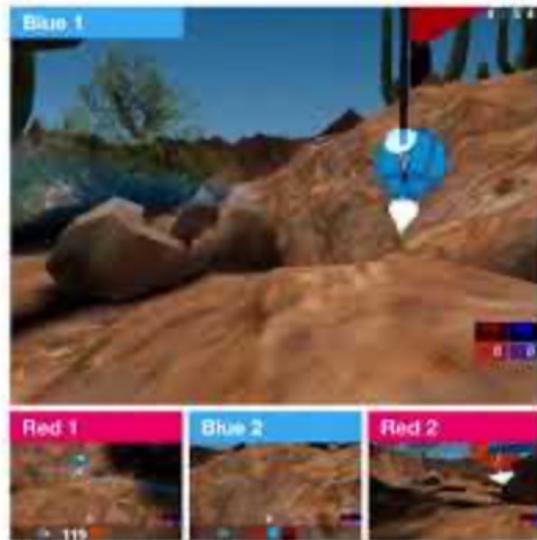
Long short-term memory networks



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

Reinforcement learning

Agent observation raw pixels



Outdoor map overview

<https://youtu.be/OjVxXyp7Bxw>

Reinforcement learning



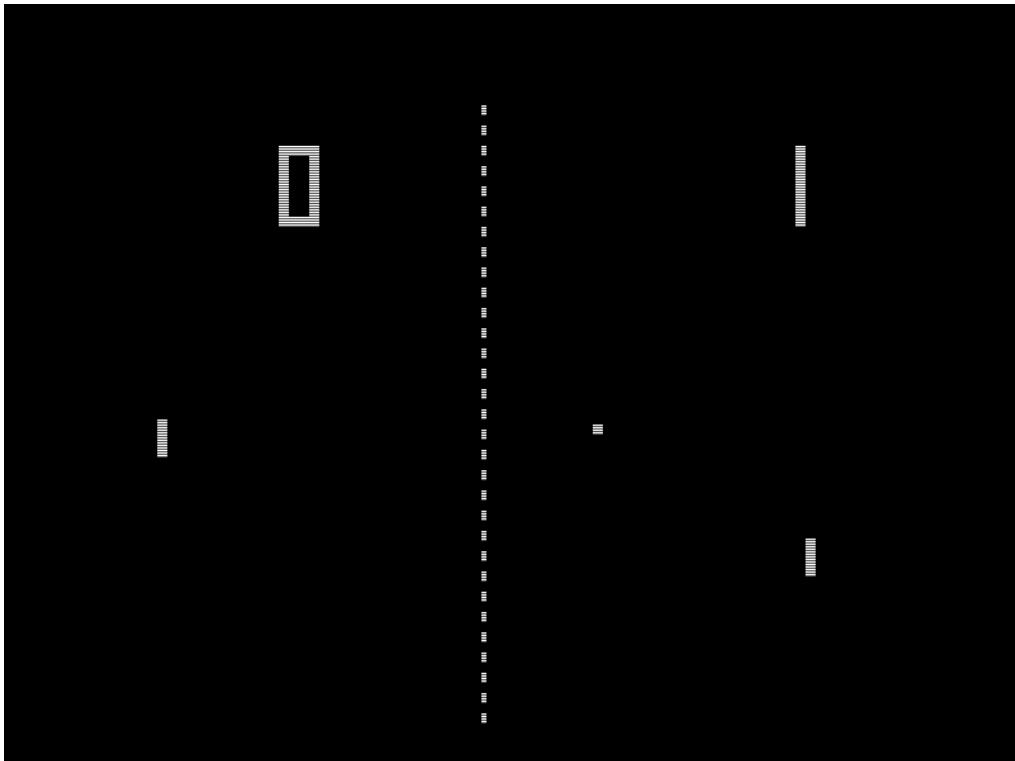
<https://youtu.be/kopoLzvh5jY>

Reinforcement learning

Reinforcement learning needs:

- A world, e.g. a game
- Agent(s)
- Self-play, i.e. millions of iterations
- Rewards & penalties

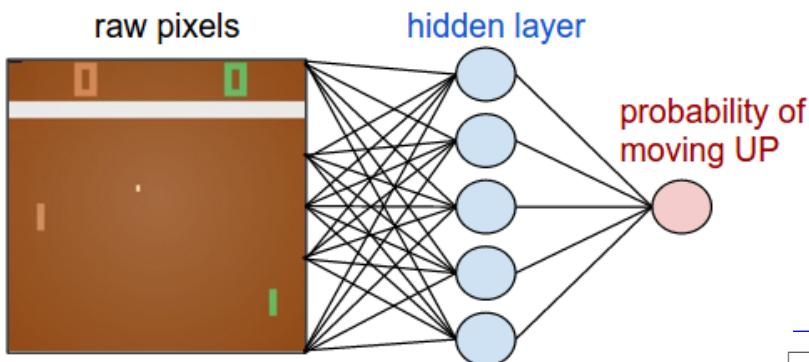
Pong



(Figure from <https://en.wikipedia.org/wiki/Pong>)

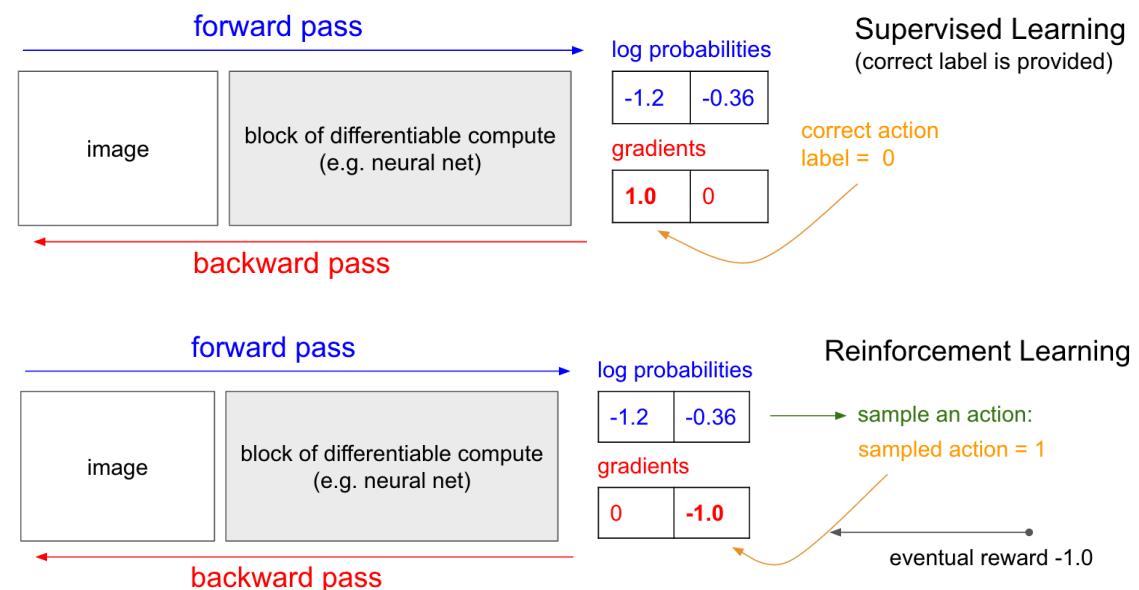


Reinforcement learning



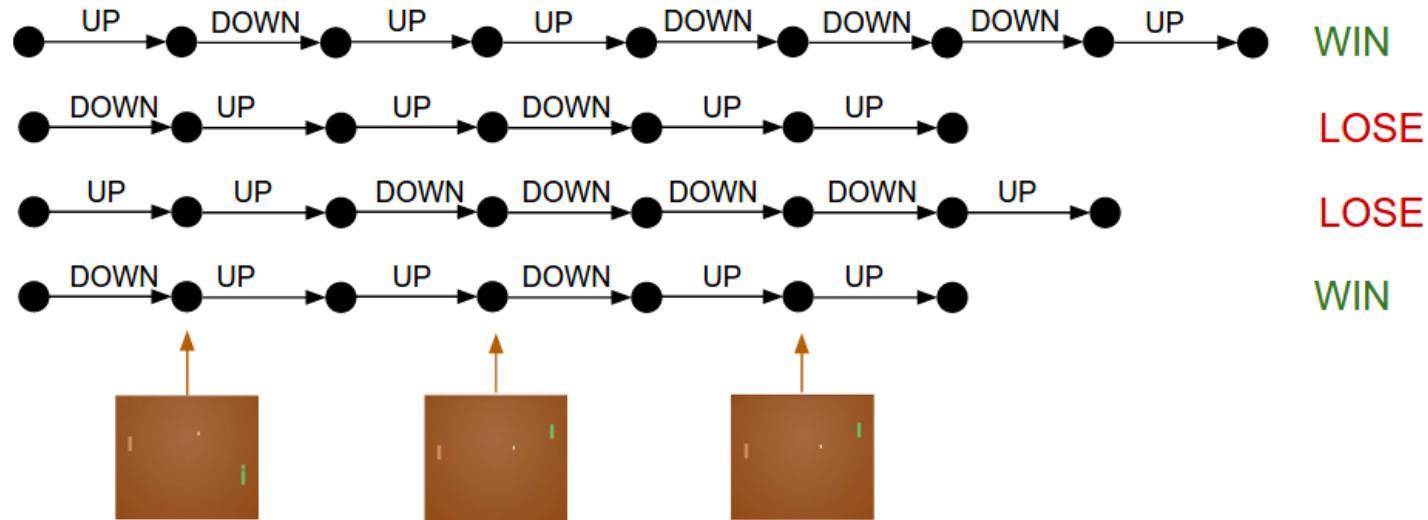
Policy network

This network will take the state of the game and decide what we (i.e. the agent) should do.



(Figure from <http://karpathy.github.io/2016/05/31/r1/>)

Reinforcement learning

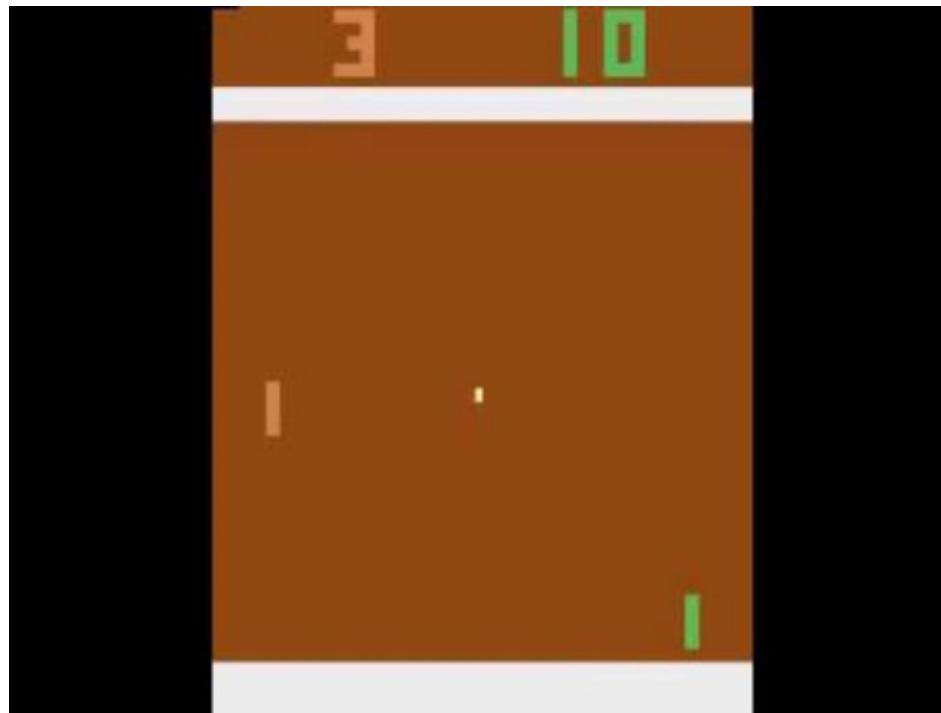


Credit assignment problem

Which actions are the actions
that make a difference

(Figure from <http://karpathy.github.io/2016/05/31/rl/>)

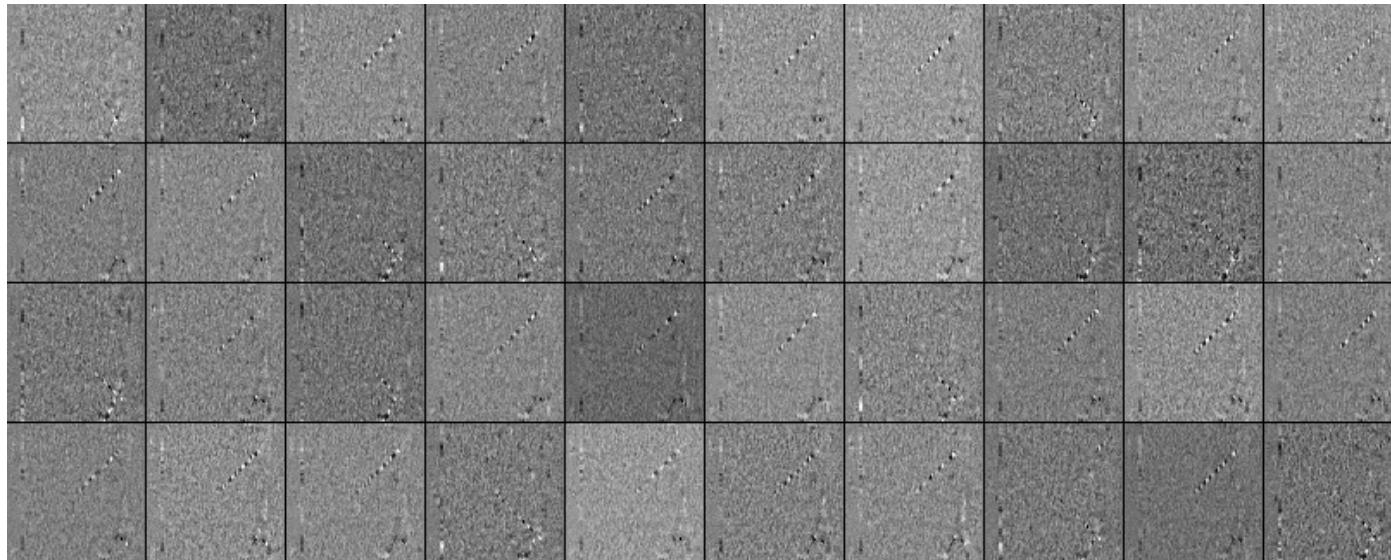
Reinforcement learning



<https://youtu.be/YOW8m2YGtRg>

(Film from <http://karpathy.github.io/2016/05/31/rl/>)

Reinforcement learning



(Figure from <http://karpathy.github.io/2016/05/31/rl/>)

Reinforcement learning

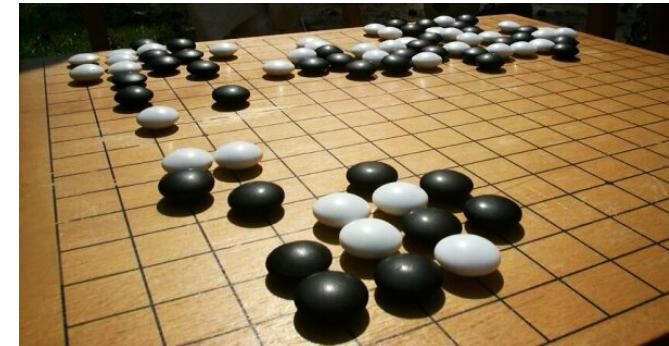
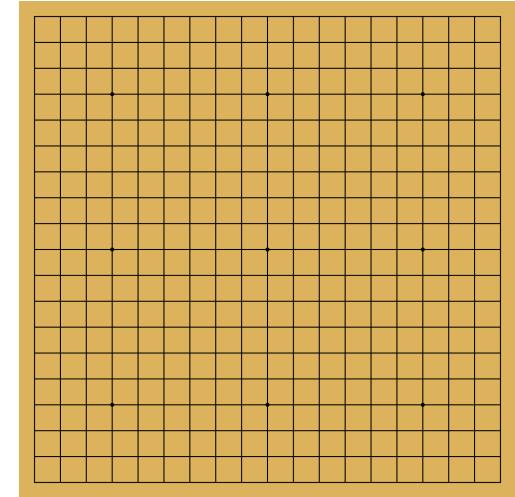
- World to act in
- Agent(s) that act in the world
- Policy network, what to do
- Self-play
- Rewards & penalties

Go

- $2.08168199382 \times 10^{170}$ possible moves
- No AI/software has beaten a professional

“The Game of Go is the holy grail of artificial intelligence. Everything we've ever tried in AI, it just falls over when you try the game of Go.”

David Silver



Figures from <https://da.wikipedia.org/wiki/Go>

Deep blue



Garry Kasparov
(1963–)



<https://en.wikipedia.org/wiki/chess>

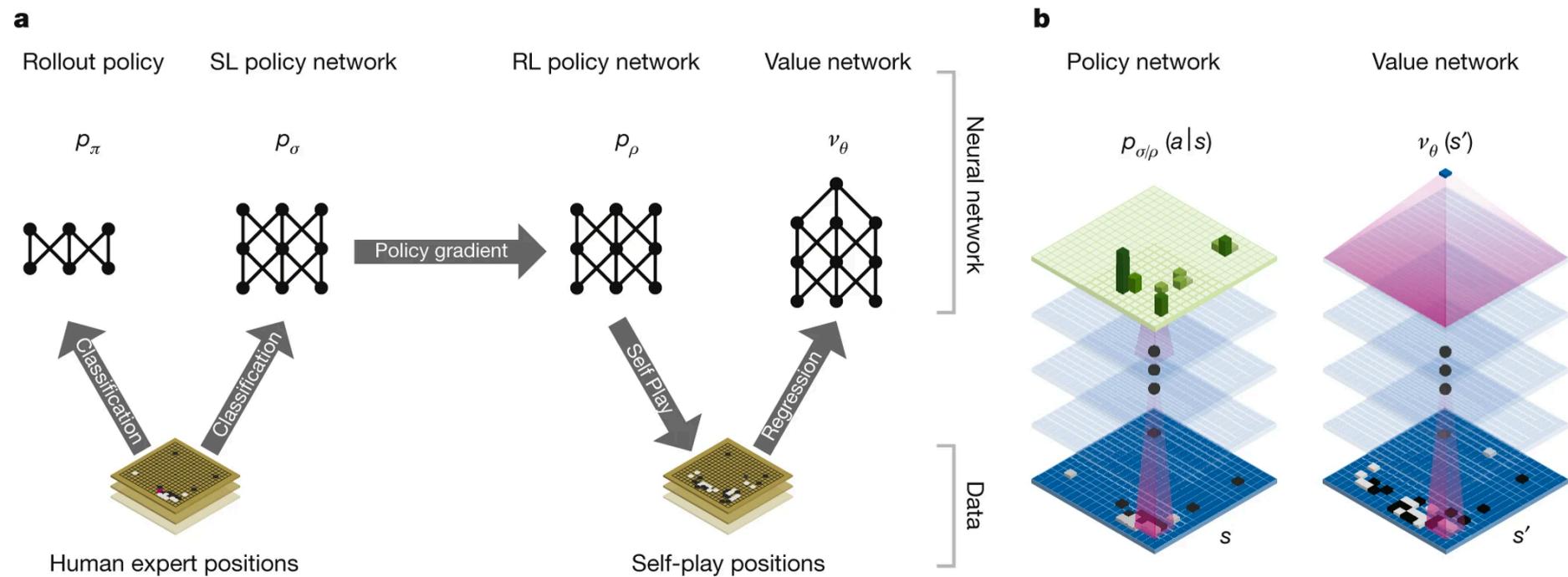


[https://en.wikipedia.org/wiki/Deep_Blue_\(chess_computer\)](https://en.wikipedia.org/wiki/Deep_Blue_(chess_computer))

AlphaGo

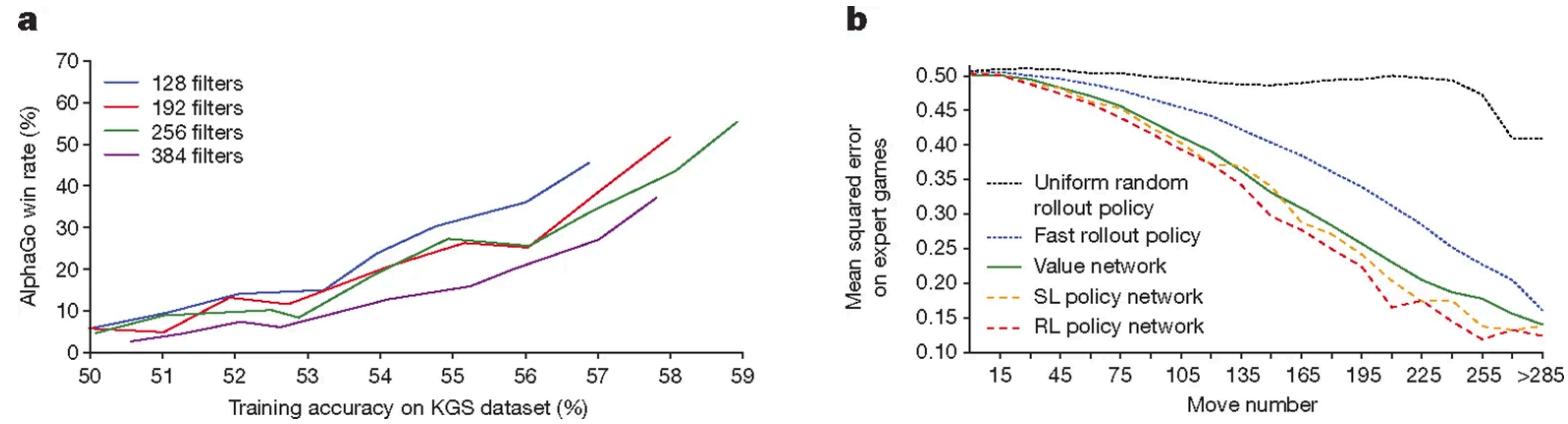


AlphaGo



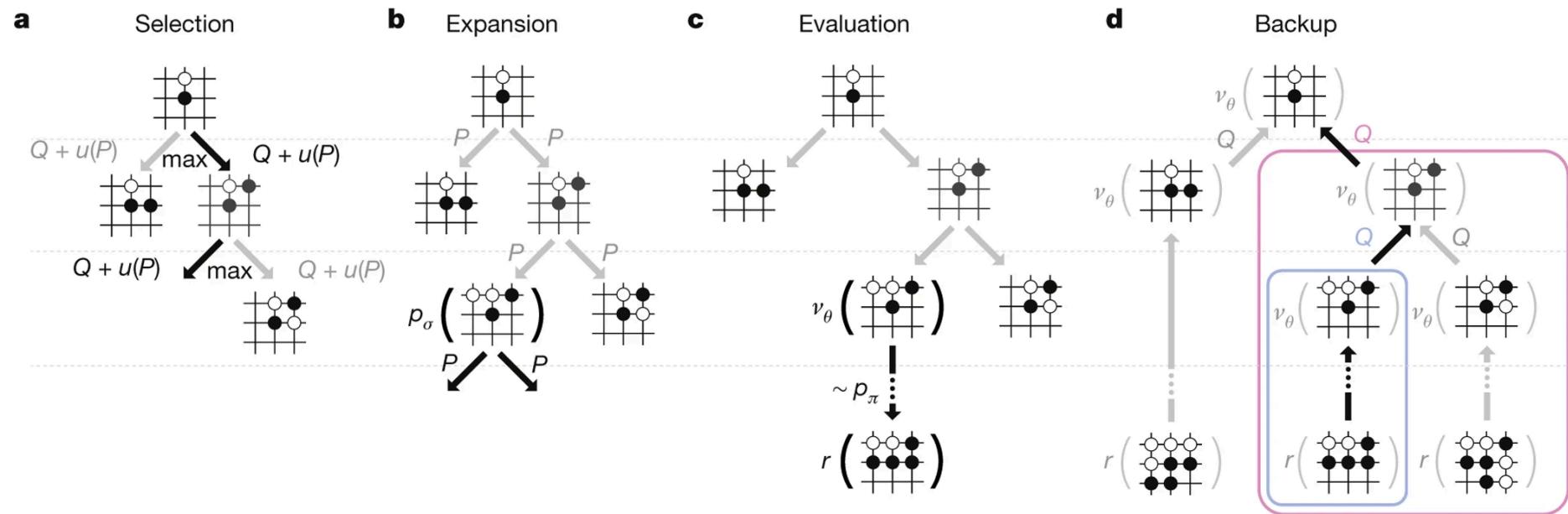
(Figure from Silver et al. 2016)

AlphaGo



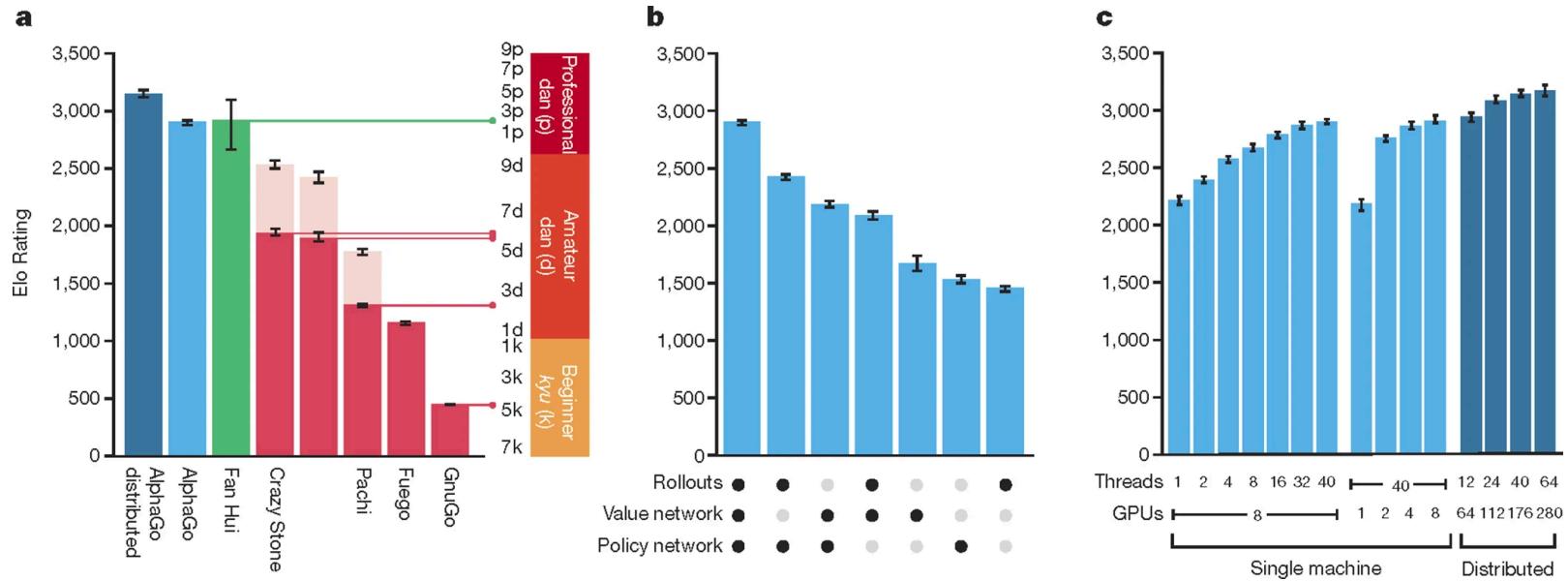
(Figure from Silver et al. 2016)

AlphaGo



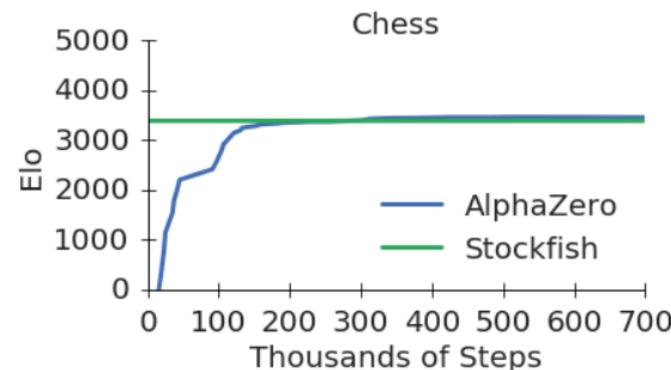
(Figure from Silver et al. 2016)

AlphaGo

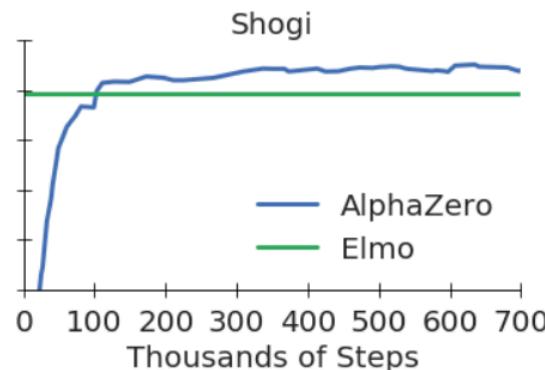


(Figure from Silver et al. 2016)

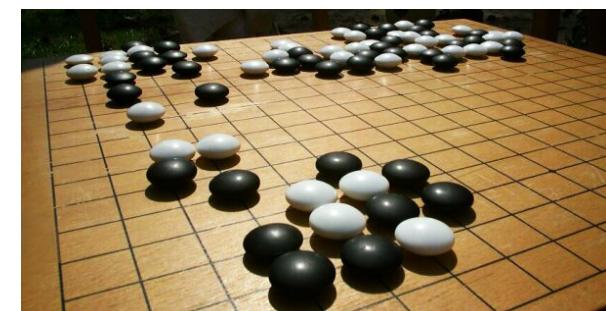
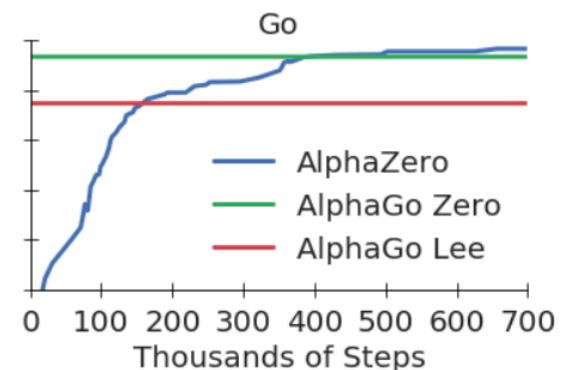
AlphaZero



<https://en.wikipedia.org/wiki/chess>



<https://en.wikipedia.org/wiki/shogi>



Figures from <https://da.wikipedia.org/wiki/Go>

(Figure from Silver et al. 2017)

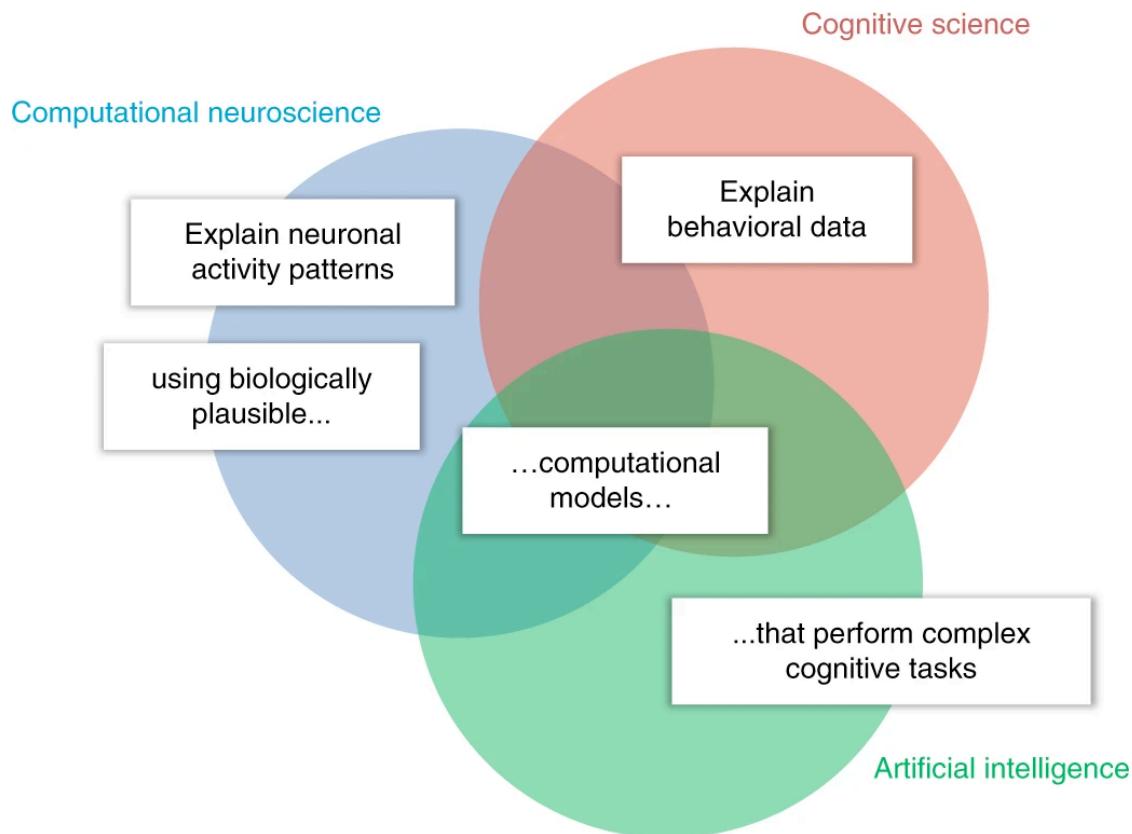
Cognitive computational neuroscience

Neural networks as models

“Understanding the brain requires that we develop theory and experiment in tandem and complement the bottom-up, data-driven approach by a top-down, theory-driven approach that starts with behavioral functions to be explained”

(Kriegeskorte & Douglas, 2018, p. 1157)

Cognitive computational neuroscience

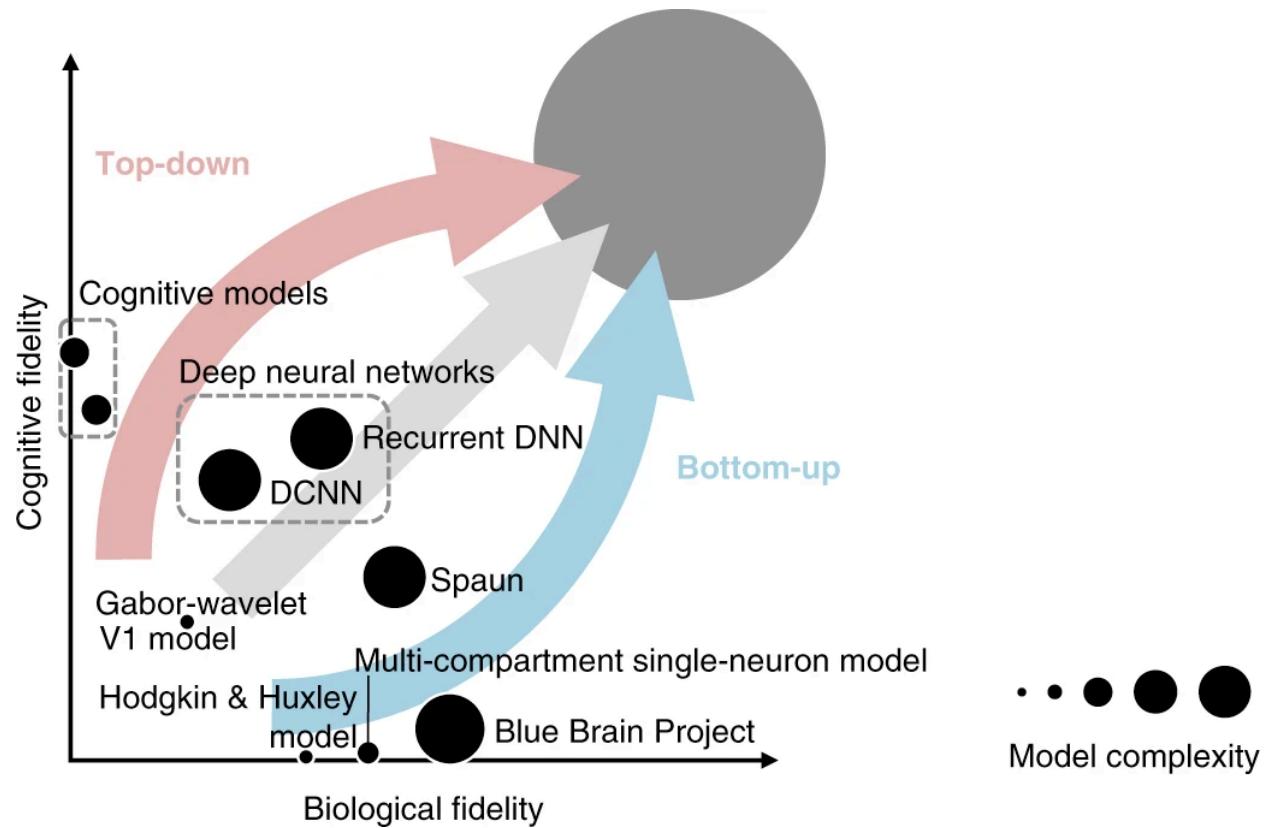


(Figures from Kriegeskorte & Douglas 2018)

Models

- Data-analysis models
- Effective connectivity; Causal-interaction models
- Box-and-arrow model
- Word model
- Oracle mode
- Brain-computational model (BCM)
- Reinforcement learning models
- Sensory encoding model
- Internal-transformation model
- Behavioral decoding model
- Psychophysical models
- Cognitive models
- Model is used to refer to models of the world employed by the brain

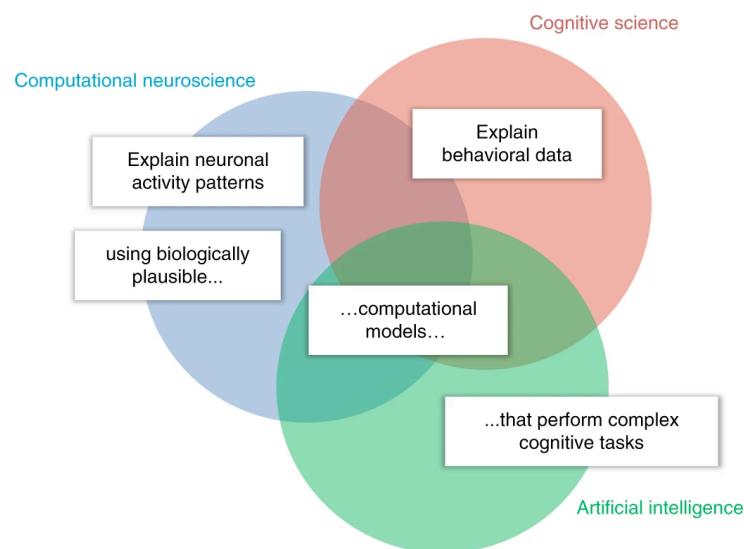
Models



(Figures from Kriegeskorte & Douglas, 2018)

Marr's levels of analysis

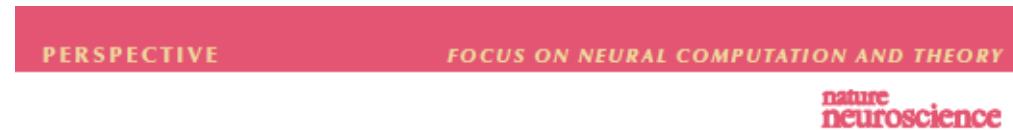
- 1) Computational theory
- 2) Representation and algorithm; and
- 3) Neurobiological implementation



- “**Cognitive science** starts from computational theory, decomposing cognition into components and developing representations and algorithms from the top down.”
- “**Computational neuroscience** proceeds from the bottom up, composing neuronal building blocks into representations and algorithms thought to be useful components in the context of the brain’s overall function.”
- “**AI** builds representations and algorithms that combine simple components to implement complex feats of intelligence.”

(Kriegeskorte & Douglas 2018, p. 1157)

Yamins & DiCarlo 2016



Using goal-driven deep learning models to understand sensory cortex

Daniel L K Yamins^{1,2} & James J DiCarlo^{1,2}

Using goal-driven deep learning models...

“The goal-driven approach is inspired by the idea that, whatever parameters are used, a neural network will have to be effective at solving the behavioral tasks the sensory system supports to be a correct model of a given sensory system.”

(Yamins & DiCarlo, 2016, p. 359)

Using goal-driven deep learning models...

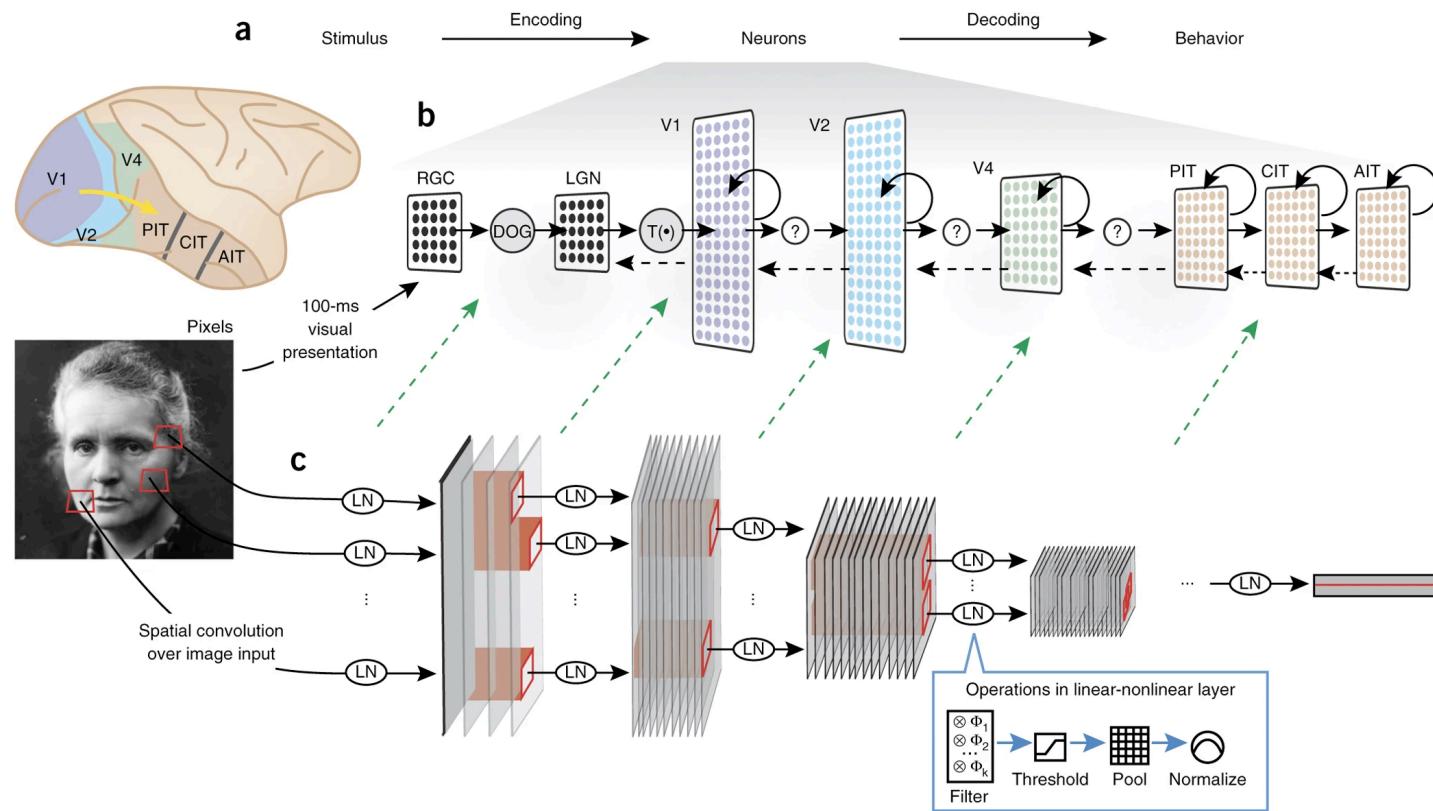


Figure from Yamins & DiCarlo (2016)

Using goal-driven deep learning models...

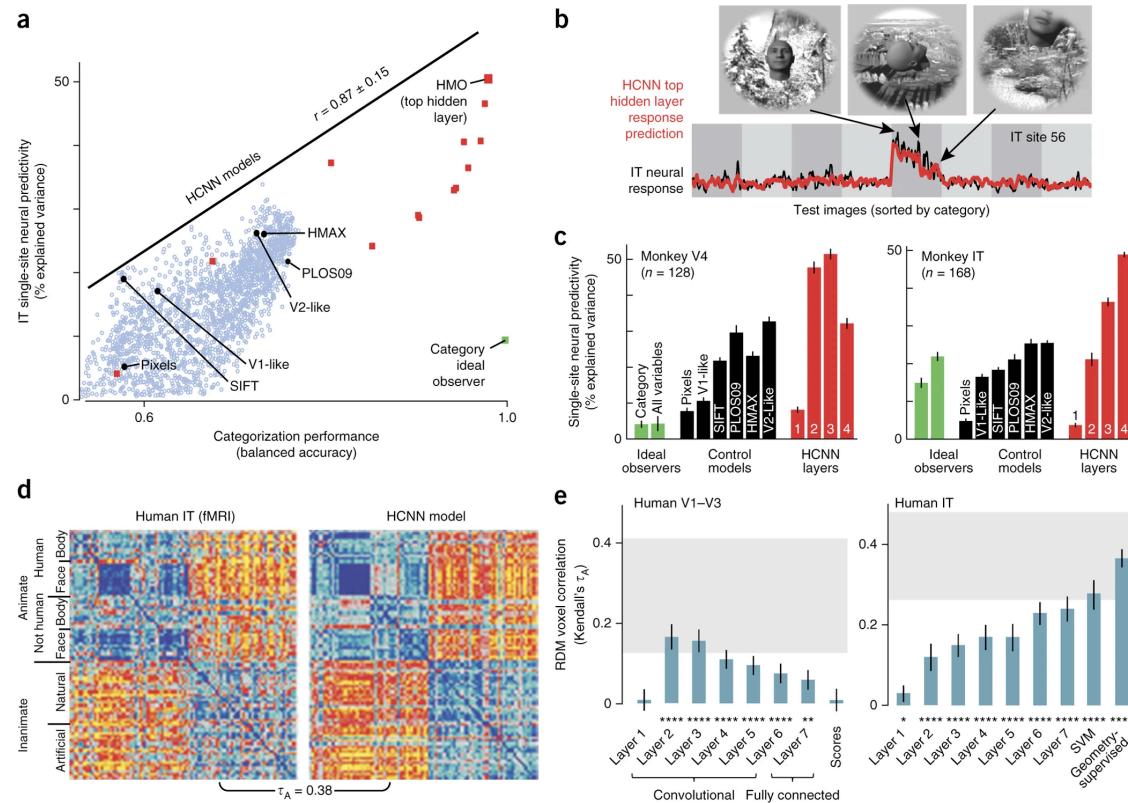


Figure from Yamins & DiCarlo (2016)

Using goal-driven deep learning models...

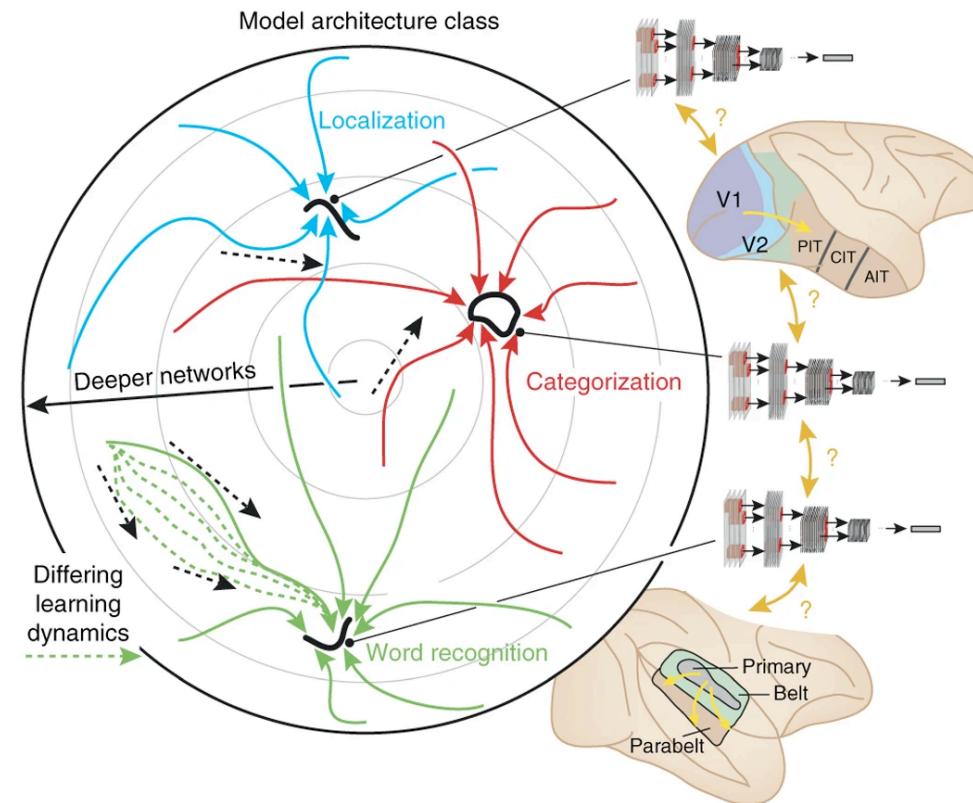


Figure from Yamins & DiCarlo (2016)

Generative Adversarial Networks

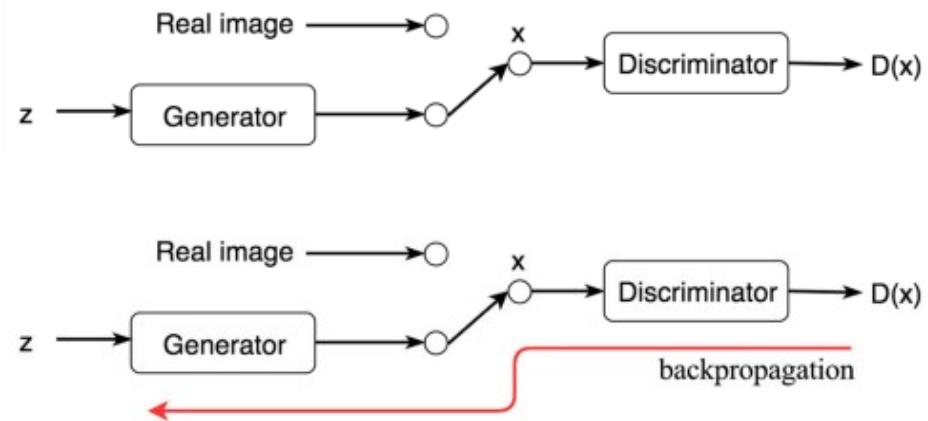
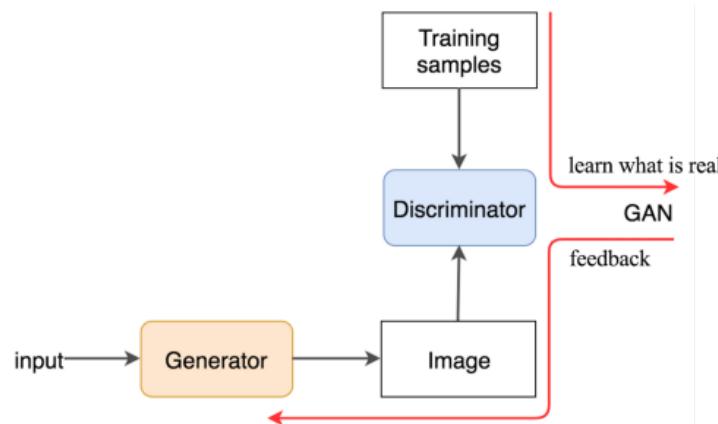
“The main focus for GAN (Generative Adversarial Networks) is to generate data from scratch”

(From *GAN – What Is Generative Adversarial Networks GAN?*, 2018)

<https://jonathan-hui.medium.com/gan-whats-generative-adversarial-networks-and-its-application-f39ed278ef09>

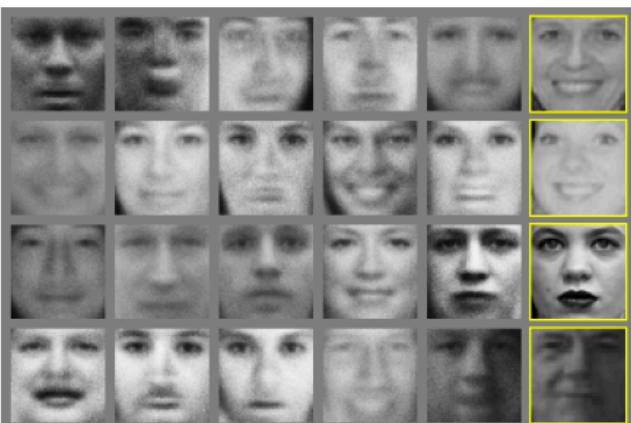
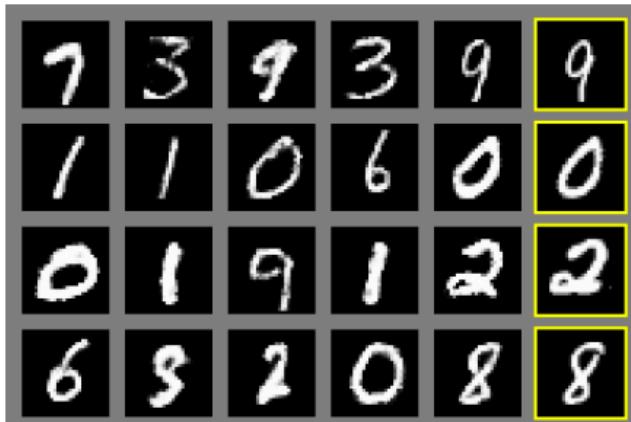


Generative Adversarial Networks



(Figures from *GAN — What Is Generative Adversarial Networks GAN?*, 2018)

Generative Adversarial Networks



(Figures from Goodfellow et al., 2014)



(Figures from Karras et al., 2018)

EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces

Vernon J Lawhern^{1,5}, Amelia J Solon^{1,2}, Nicholas R Waytowich^{1,3},
Stephen M Gordon^{1,2}, Chou P Hung^{1,4} and Brent J Lance¹

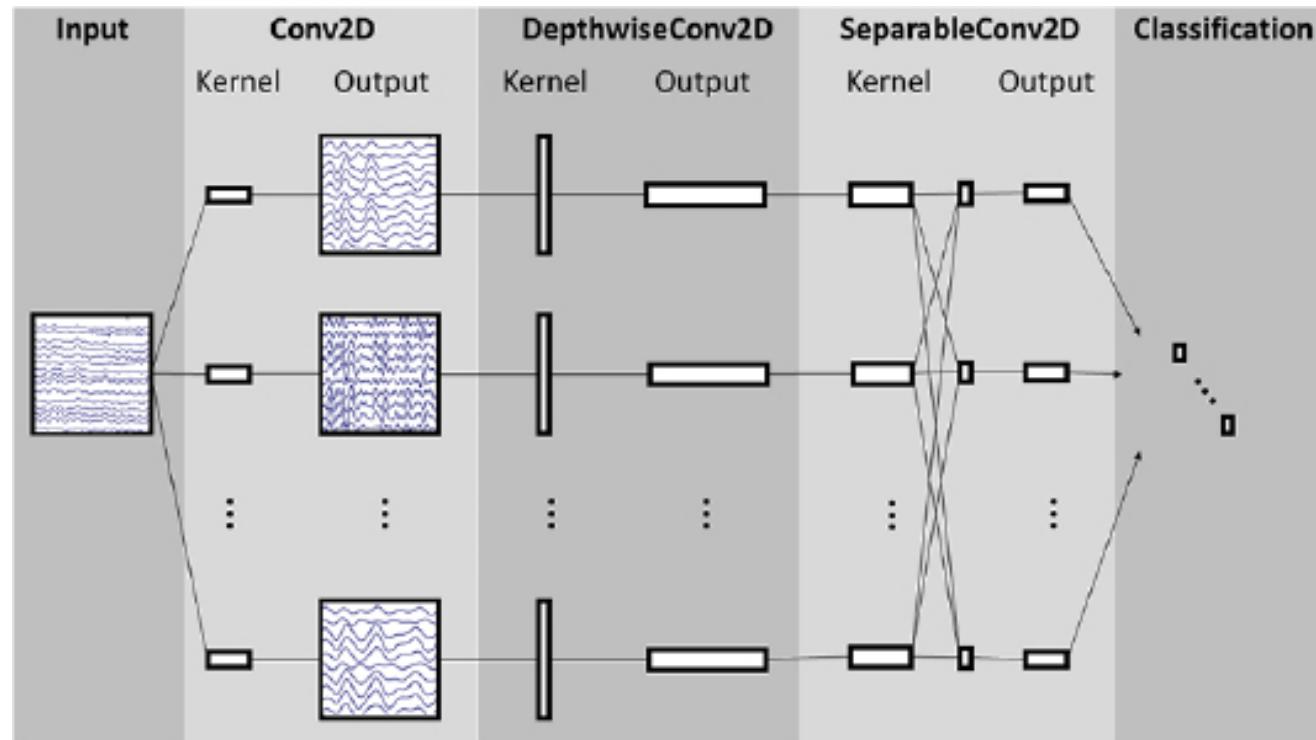
¹ Human Research and Engineering Directorate, U.S. Army Research Laboratory, Aberdeen Proving Ground, MD, United States of America

² DCS Corporation, Alexandria, VA, United States of America

³ Department of Biomedical Engineering, Columbia University, New York, NY, United States of America

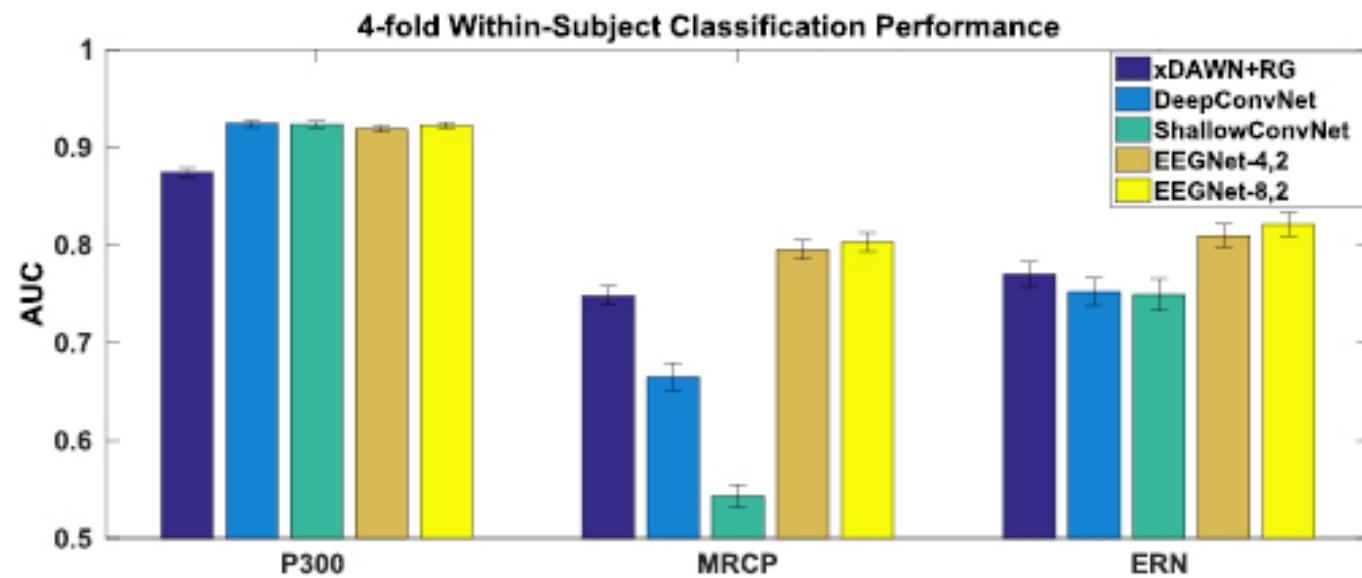
⁴ Department of Neuroscience, Georgetown University, Washington, DC, United States of America

Neural network for EEG



(Figure from Lawhern et al., 2016)

Neural network for EEG



(Figure from Lawhern et al., 2016)

Topic(s) for last lecture?

Questions?

1. Recurrent neural networks
2. Reinforcement learning
 1. Pong
 2. AlphaGo
3. Networks as models
4. Generative Adversarial Networks (GANs)
5. Neural network for EEG
6. Last lecturez

References

- Andrej Karpathy. (2016, May 31). Deep Reinforcement Learning: Pong from Pixels. <http://karpathy.github.io/2016/05/31/rl/>
- GAN – What is Generative Adversarial Networks GAN? (2018, June 19). <https://jonathan-hui.medium.com/gan-whats-generative-adversarial-networks-and-its-application-f39ed278ef09>
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative Adversarial Networks. *ArXiv:1406.2661 [Cs, Stat]*. <http://arxiv.org/abs/1406.2661>
- Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2018). Progressive Growing of GANs for Improved Quality, Stability, and Variation. *ArXiv:1710.10196 [Cs, Stat]*. <http://arxiv.org/abs/1710.10196>
- Kriegeskorte, N., & Douglas, P. K. (2018). Cognitive computational neuroscience. *Nat Neurosci*, 21(9), 1148–1160. <https://doi.org/10.1038/s41593-018-0210-5>
- Lawhern, V. J., Solon, A. J., Waytowich, N. R., Gordon, S. M., Hung, C. P., & Lance, B. J. (2016). EEGNet: A Compact Convolutional Network for EEG-based Brain-Computer Interfaces. *ArXiv E-Prints*. <https://doi.org/10.1088/1741-2552/aace8c>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436–444. <https://doi.org/10.1038/nature14539>
- Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587), 484–489. <https://doi.org/10.1038/nature16961>
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K., & Hassabis, D. (2017). Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. *ArXiv:1712.01815 [Cs]*. <http://arxiv.org/abs/1712.01815>
- Yamins, D. L. K., & DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3), 356–365. <https://doi.org/10.1038/nn.4244>
- Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2020). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *ArXiv:1703.10593 [Cs]*. <http://arxiv.org/abs/1703.10593>