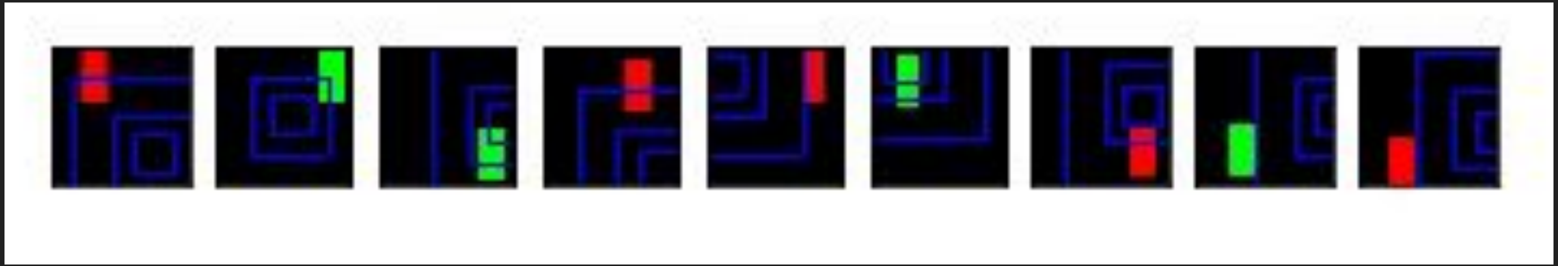
Building and Replicating Models of Visual Search Behavior



with Tensorflow and the Scientific Python Stack

David Nicholson
Emory University, Biology, Prinz lab

NickleDave   @nicholdav

Acknowledgements

- Atlanteans

Constantine
Dovrolis

Zsolt
Kira

Sarah
Pallas

Astrid
Prinz

- DARPA Lifelong Learning Machines (L2M) program

Acknowledgements

pygame

pandas

Joblib

imageio

Introduction

Visual search:

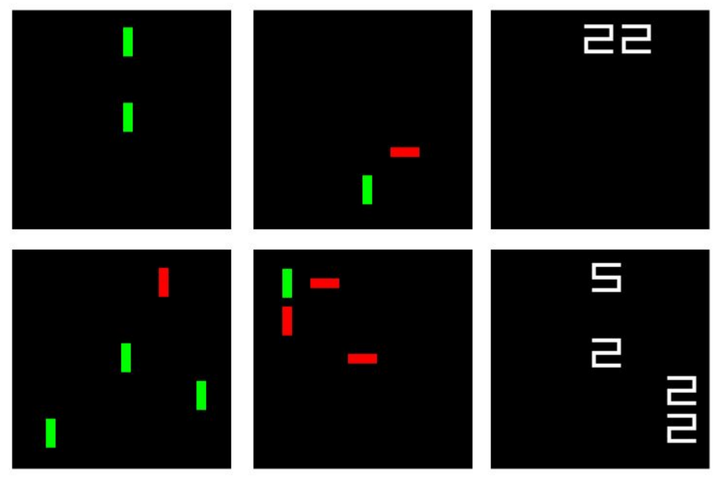
- in the real world

http://info.ni.tu-berlin.de/photodb/

Introduction

Visual search:

- in the real world
- in the lab



Introduction

Why build models of visual search **behavior**?

1. understand brain mechanisms of **goal-driven perception**

Introduction

Why build models of visual search **behavior**?

1. understand brain mechanisms of **goal-driven perception**
    - Does the model we build with this mechanism **behave** like humans and other animals?

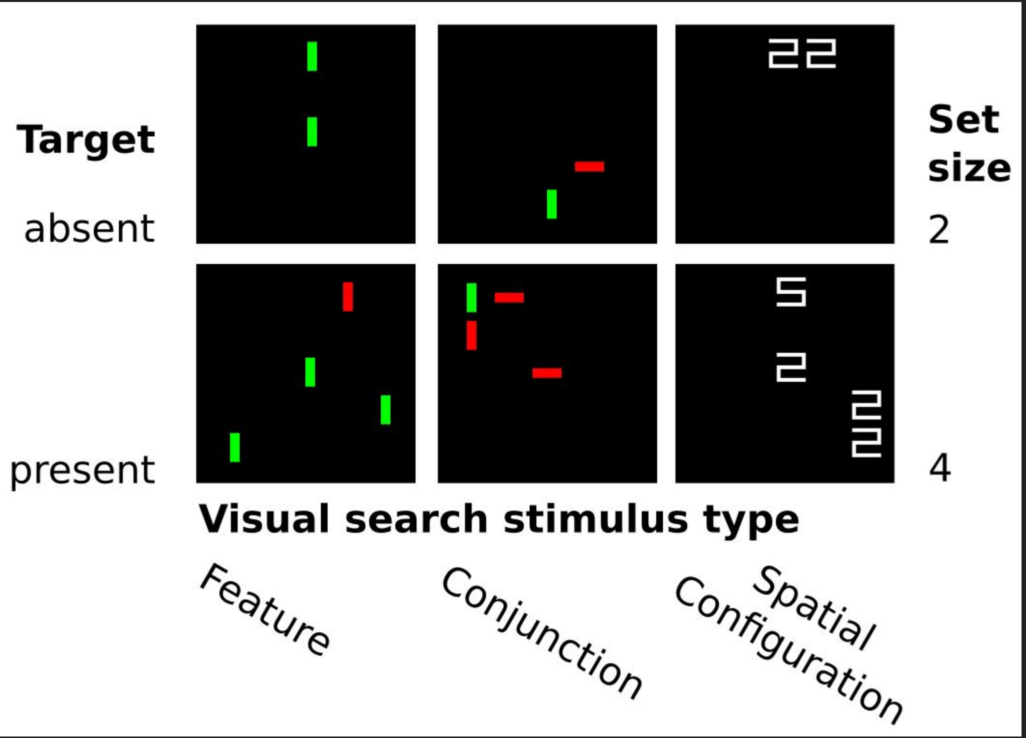
Introduction

Why build models of visual search **behavior**?

1. understand brain mechanisms of **goal-driven perception**
    - Does the model we build with this mechanism **behave** like humans and other animals?
2. design artificial intelligence algorithms that draw from these mechanisms
    - Does our agent **behave** like humans and other animals?

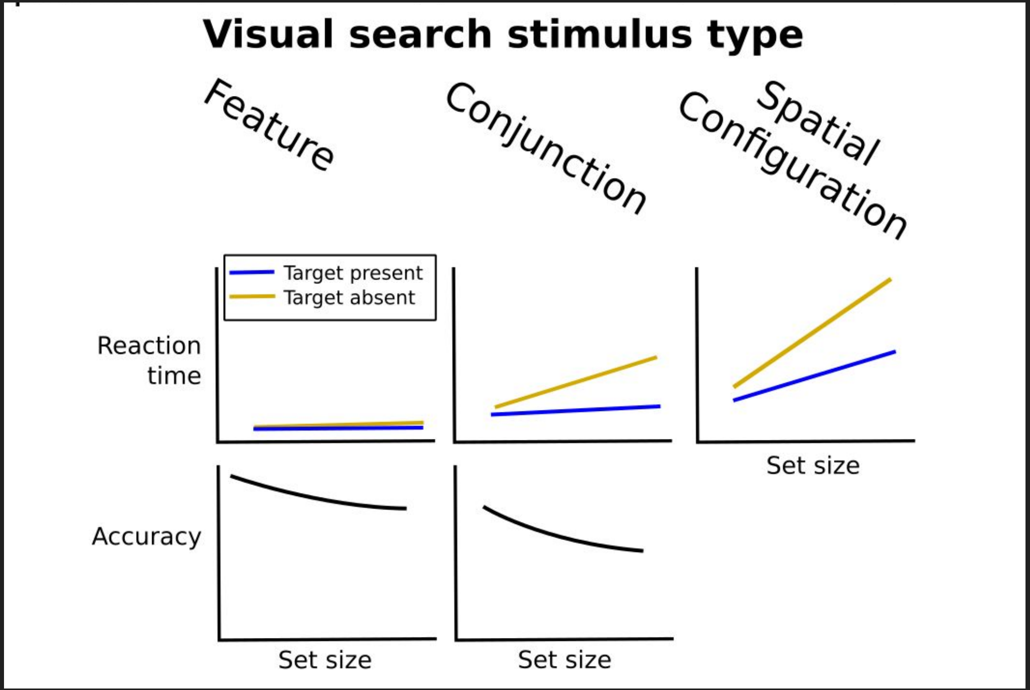
Introduction

The *discrete item display visual search* task



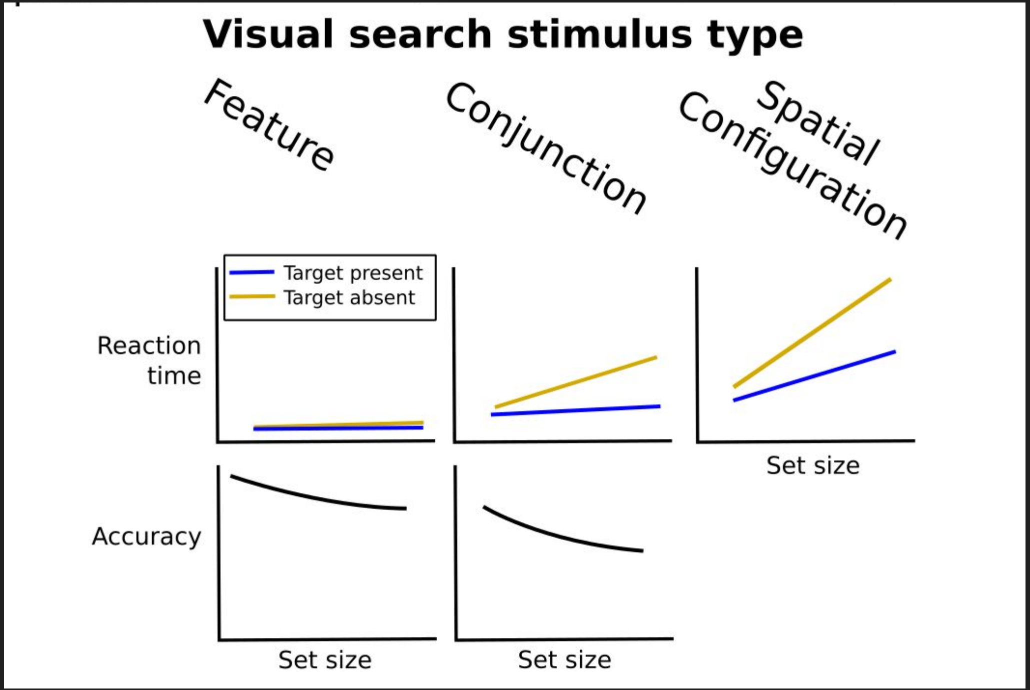
Introduction

The *discrete item display visual search* task



Introduction

***Models* of the *discrete item display visual search* task**



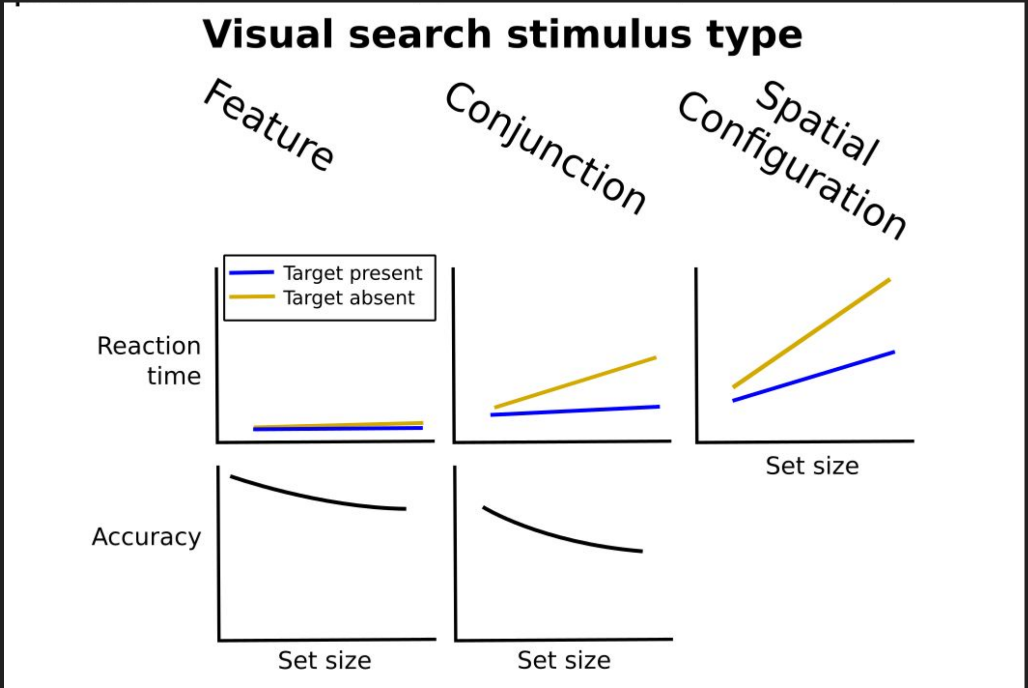
Models of
capacity limitations

serial, attention-limited
- e.g. Guided Search

parallel, noise-limited
- Signal Detection Theory-based models

Introduction

***Models* of the *discrete item display visual search* task**



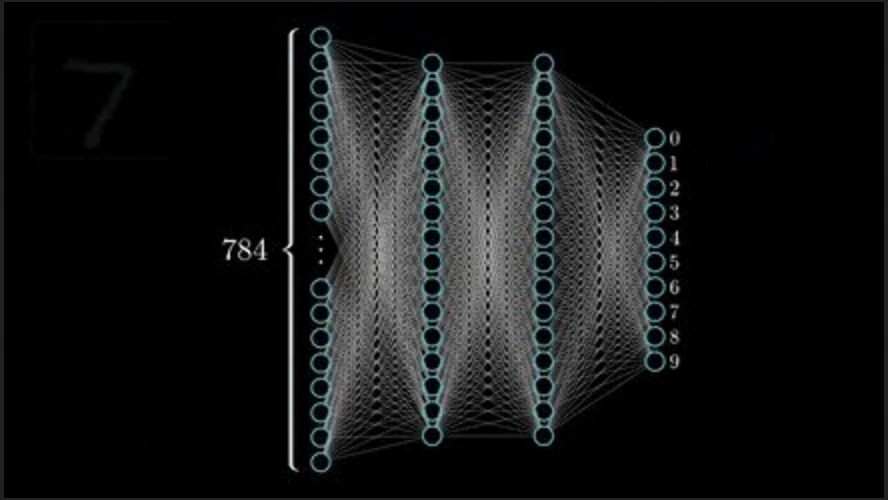
Models of
capacity limitations

None of these models are
"pixels-in, behavior out"

Introduction

Neural networks as models of the visual system

What I mean by "neural networks"



https://www.youtube.com/watch?v=aircAruvnKk

Introduction

Neural networks as models of the visual system

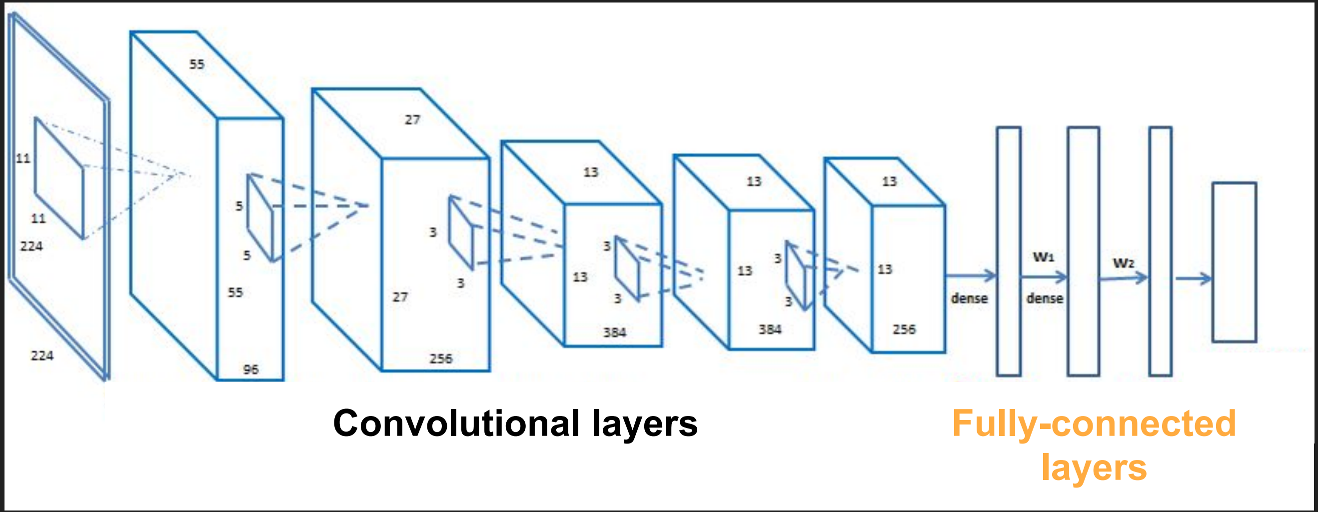
Specifically, **convolutional neural networks (CNNs)**

https://cs.nyu.edu/~fergus/tutorials/deep_learning_cvpr12/

Introduction

Neural networks as models of the visual system

The architecture of CNNs resembles the visual system

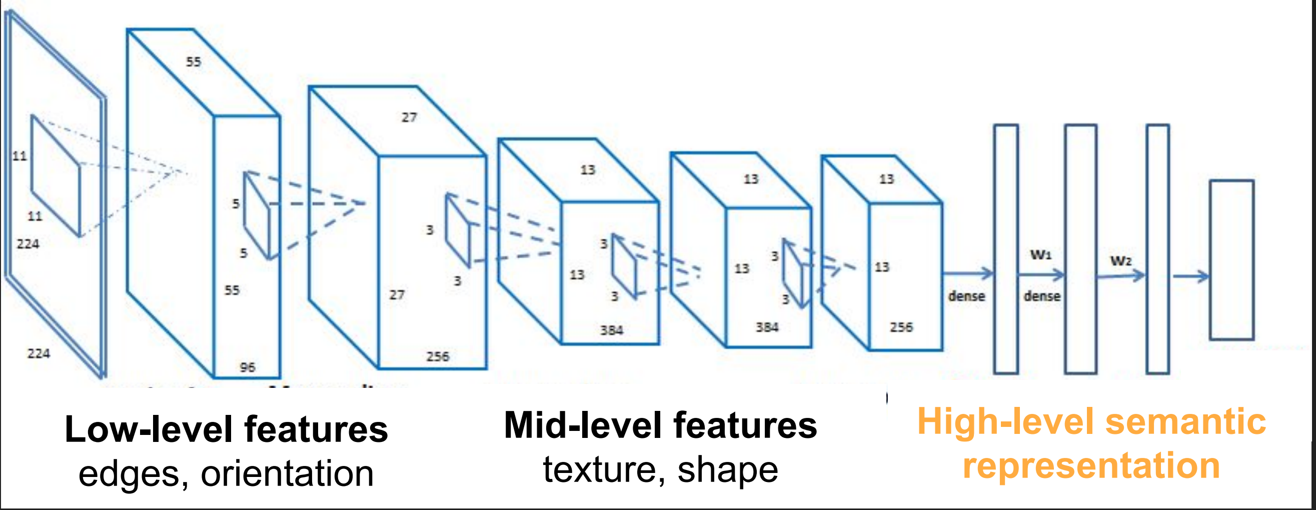


AlexNet (Krizhevsky et al. 2012). Adapted from Wang et al. 2015

Introduction

Neural networks as models of the visual system

The architecture of CNNs resembles the visual system

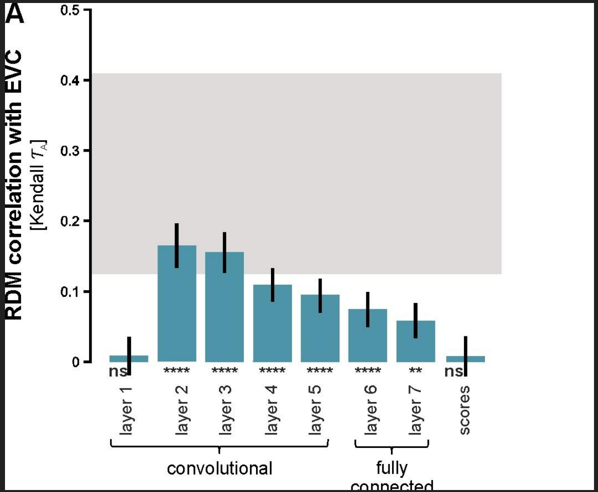


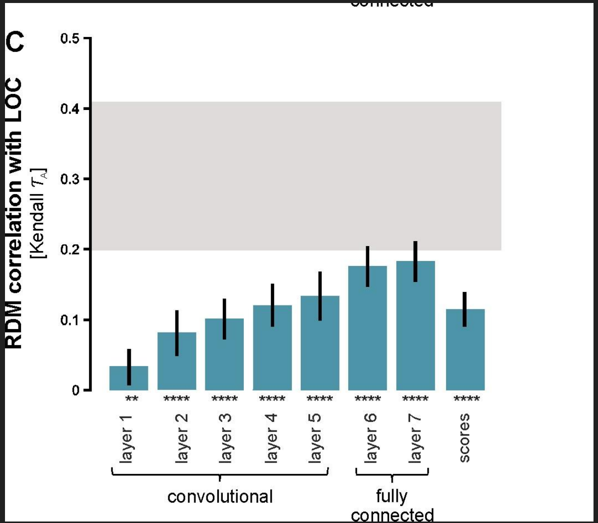
"the (primate) visual system"

Introduction

Neural networks as models of the visual system

Task-optimized CNNs learn representations similar to those in the visual system of the (primate) brain



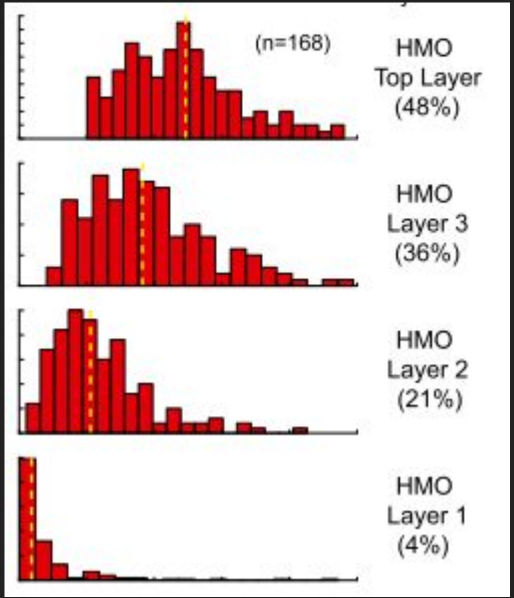


Khaligh-Razavi, Kriegeskorte 2014

Introduction

Neural networks as models of the visual system

Task-optimized CNNs learn representations similar to those in the visual system of the (primate) brain



Yamins DiCarlo 2014

Introduction

Neural networks as brain models

Hypothesis: if CNNs are using representations similar to those in the (primate) visual system, then they should **behave** like humans when performing related tasks

Introduction

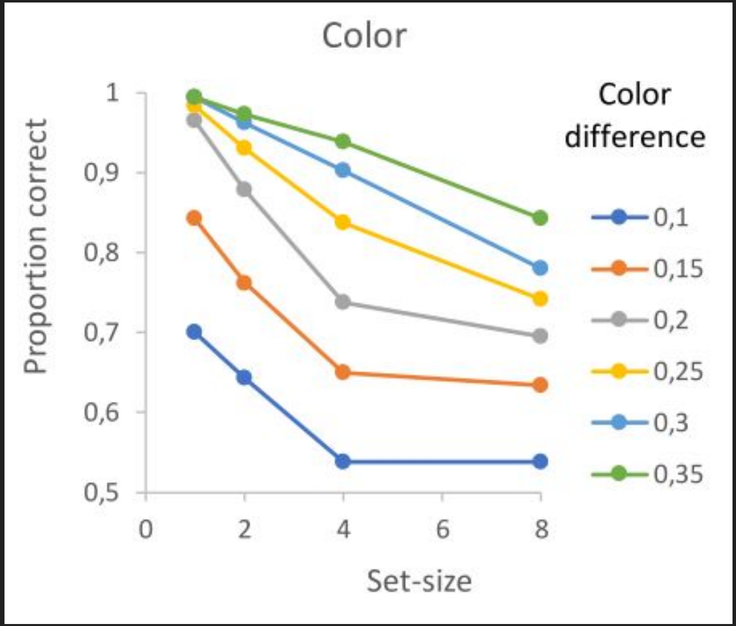
Neural networks as brain models

Hypothesis: if CNNs are using representations similar to those in the (primate) visual system, then they should **behave** like humans when performing the **discrete item display search task**

Introduction

Neural networks as brain models

Hypothesis: if CNNs are using representations similar to those in the (primate) visual system, then they should **behave** like humans when performing the **discrete item display search task**



Põder, 2017. "Capacity limitations of visual search in deep convolutional neural network"

Experiment 1: replicate fine-tuning approach

Methods

- use AlexNet and VGG16 architecture with **weights pre-trained on ImageNet dataset**

Experiment 1: replicate fine-tuning approach

Methods

- use AlexNet and VGG16 architecture with **weights pre-trained on ImageNet dataset**
- randomly initialize **fully-connected layers**

Experiment 1: replicate fine-tuning approach

Methods

- use AlexNet and VGG16 architecture with **weights pre-trained on ImageNet dataset**
- randomly initialize **fully-connected layers**
- train **fully-connected layers** with small learning rate: `0.0001`

Experiment 1: replicate fine-tuning approach

Methods

- use AlexNet and VGG16 architecture with **weights pre-trained on ImageNet dataset**
- randomly initialize **fully-connected layers**
- train **fully-connected layers** with small learning rate: `0.0001`
- apply base learning rate to other layers: `1e-20`

Experiment 1: replicate fine-tuning approach

Methods

- use AlexNet and VGG16 architecture with **weights pre-trained on ImageNet dataset**
- randomly initialize **fully-connected layers**
- train **fully-connected layers** with small learning rate: `0.0001`
- apply base learning rate to other layers: `1e-20`
- generate visual search stimuli with `searchstims`, a Python package built with PyGame (https://github.com/NickleDave/searchstims)

Experiment 1: replicate fine-tuning approach

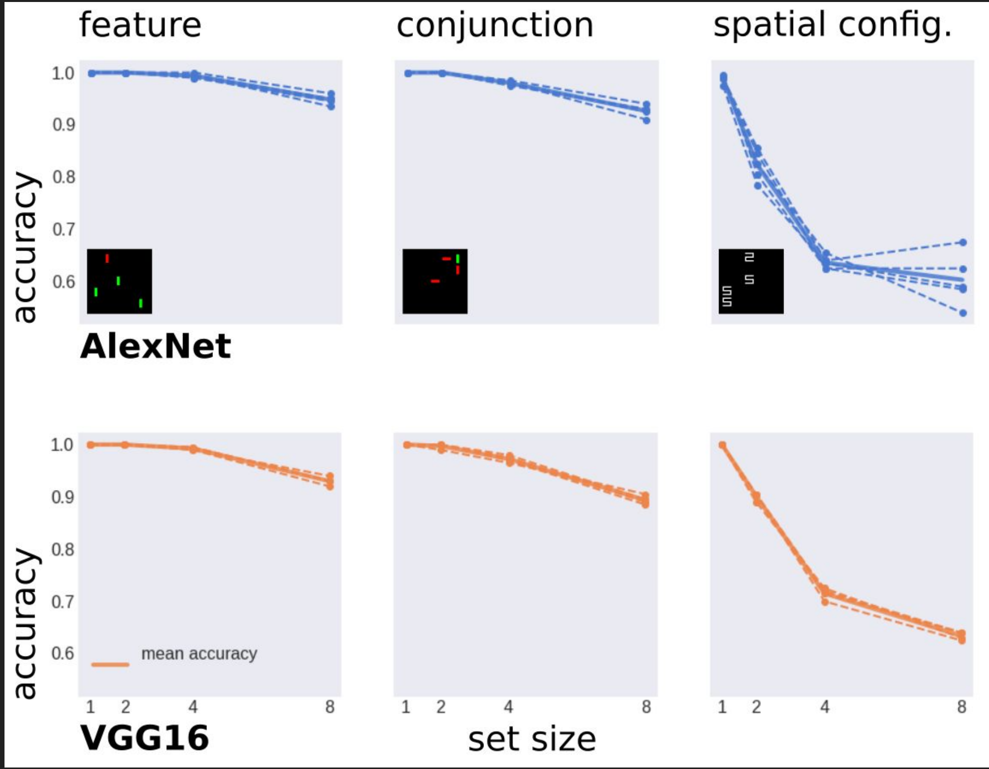
Methods

- use AlexNet and VGG16 architecture with **weights pre-trained on ImageNet dataset**
- randomly initialize **fully-connected layers**
- train **fully-connected layers** with small learning rate: `0.0001`
- apply base learning rate to other layers: `1e-20`
- generate visual search stimuli with `searchstims`, a Python package built with PyGame (https://github.com/NickleDave/searchstims)
- train 5 replicates of each network on a dataset with 6400 samples of a **single visual search stimulus**, balanced across "set size"
- measure accuracy on separate 800 sample test set

Experiment 1: replicate fine-tuning approach

Results

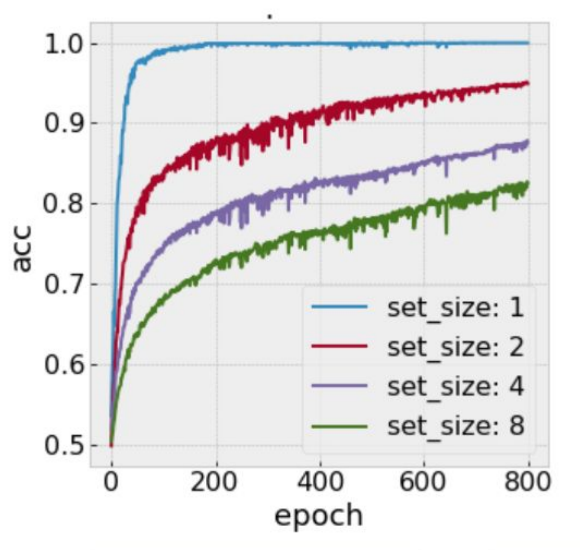
- Both AlexNet and VGG16 show human-like drop in accuracy when trained this way



Experiment 1: replicate fine-tuning approach

Results

- Both AlexNet and VGG16 show human-like drop in accuracy when trained this way
- **but** training histories suggest model has not converged

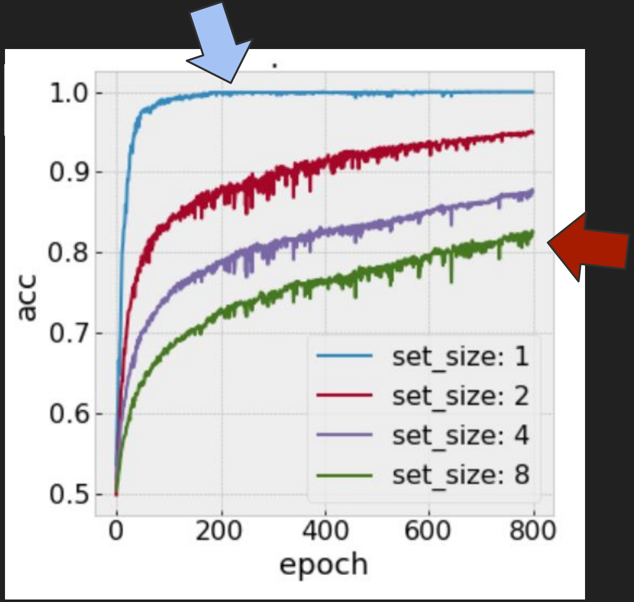


Experiment 1: replicate fine-tuning approach

Results

- notice **different rates of convergence** across set sizes:

  1 > 2 > 4 > 8



Experiment 2: typical learning rate, augment data

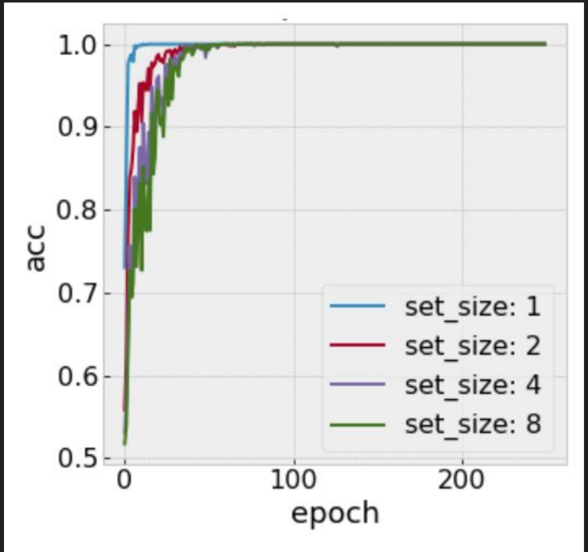
Methods

- train **fully-connected layers** with typical learning rate: `0.001`
- **freeze** weights in other layers pre-trained on ImageNet; no "base" rate
- **increase** number of training examples for larger set sizes

Experiment 2: typical learning rate, augment data

Results

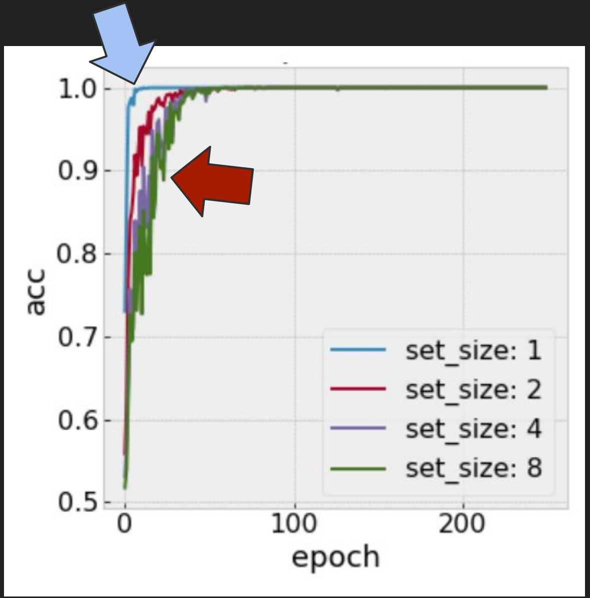
- training histories show that accuracy of models now converge on asymptotic value



Experiment 2: typical learning rate, augment data

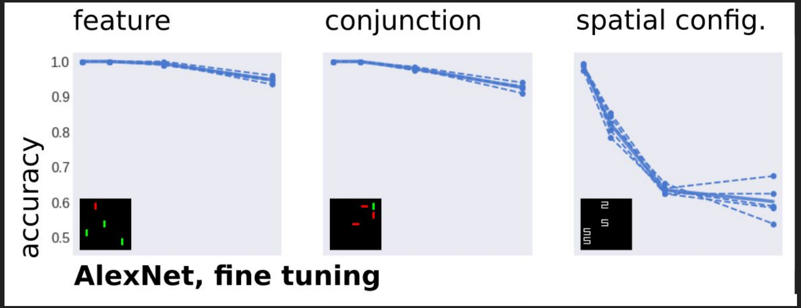
Results

- training histories show that accuracy of models now converge on asymptotic value
- but still see different rates of convergence



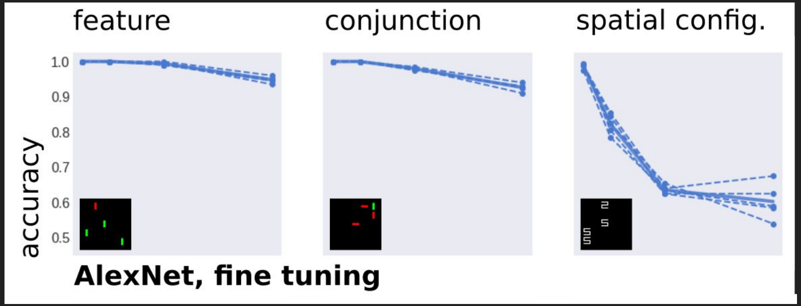
Experiment 2

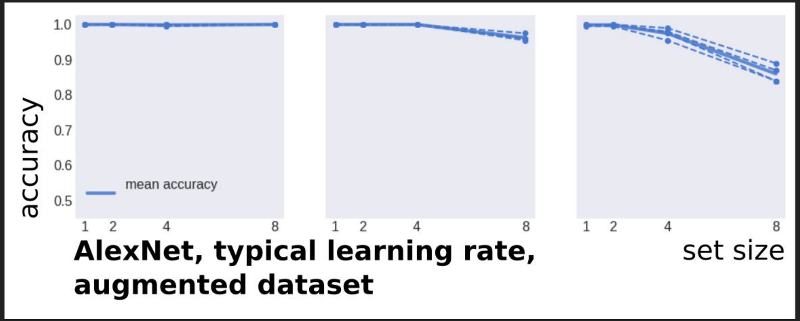
Results



Experiment 2

Results

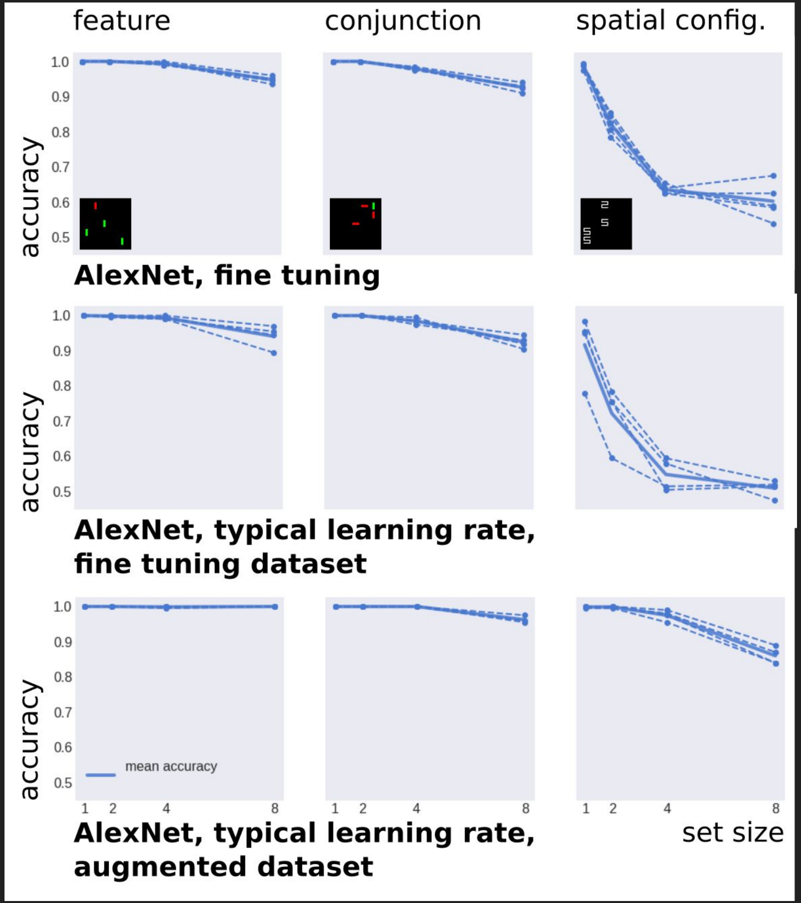




Experiment 2

Results:

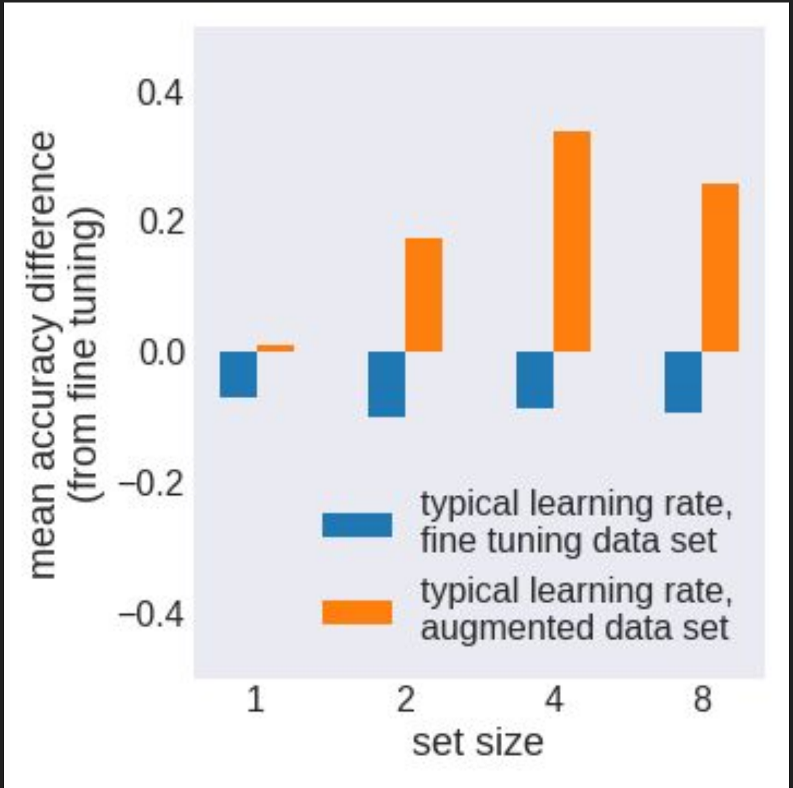
- improvement comes from augmented data



Experiment 2: typical learning rate, augment data

Results:

- improvement comes from augmented data



Discussion

implications for artificial intelligence:

- **translational invariance** is still an issue
- possible solutions:
  - spatial transformer networks (Jaderberg 2015)
  - **dynamic routing** with capsules (e.g. Sabour et al. 2017)
- are these mechanisms competitive with just augmenting the dataset?

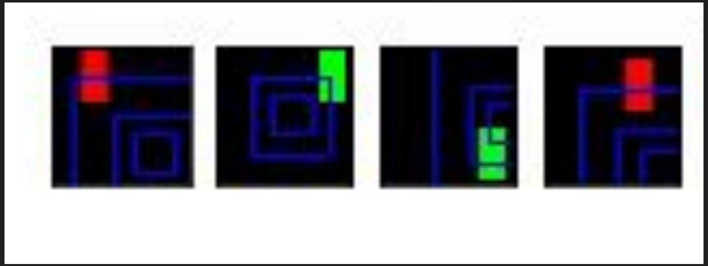
Discussion

Implications for neuroscience

- "training" the visual system may include "augmentation" to induce translational invariance
  - e.g. see just a few objects but from many different perspectives
  - cf. work by Linda Smith et al.
- visual system has other mechanisms to enable translational invariance
  - such as: moving the eyes
- hard to compare behavior of deep learning models with behavior of animals when tasks measure factors that **impair** performance
  - do we have a good model or just bad training?
  - but this is important to do; can't ignore tasks with clear effects

Questions, comments

please check out:
https://github.com/NickleDave/thrillington



https://www.nengo.ai/
https://www.nengo.ai/nengo-dl/

for more work like this, check out this conference:
https://ccneuro.org/2019/
and these podcasts
https://braininspired.co/
http://unsupervisedthinkingpodcast.blogspot.com/