Unsupervised and self-supervised learning

Lectures and outline by: Blake Richards & Timothy Lillicrap

neuromatch academy

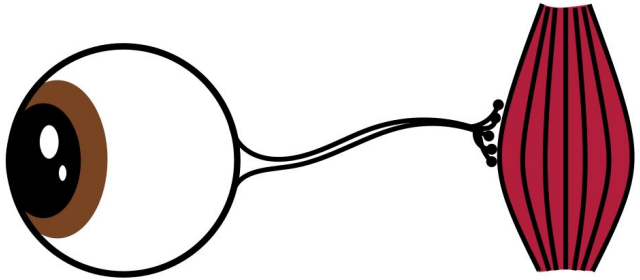
Lecture code/exercises by:
Colleen Gillon & Arna Ghosh

Representation learning without labels

Why are representations important?

This might seem a funny question, but ask yourself:
Why do you have a brain?

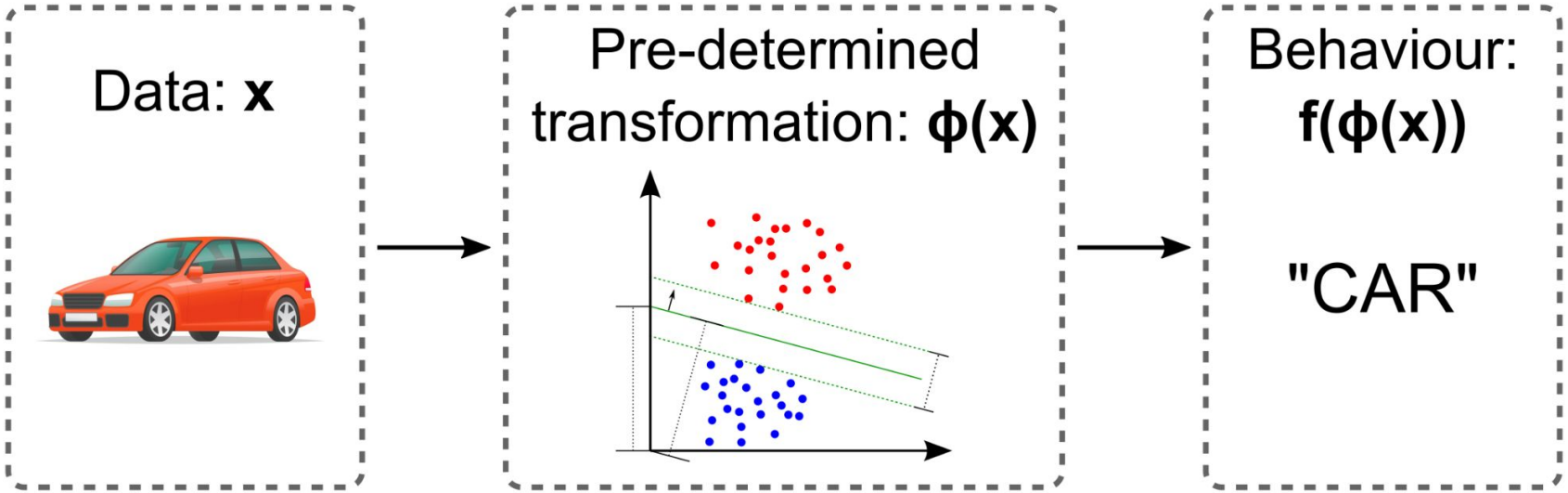
- ***Why didn't evolution just connect your sensors directly to your muscles?***



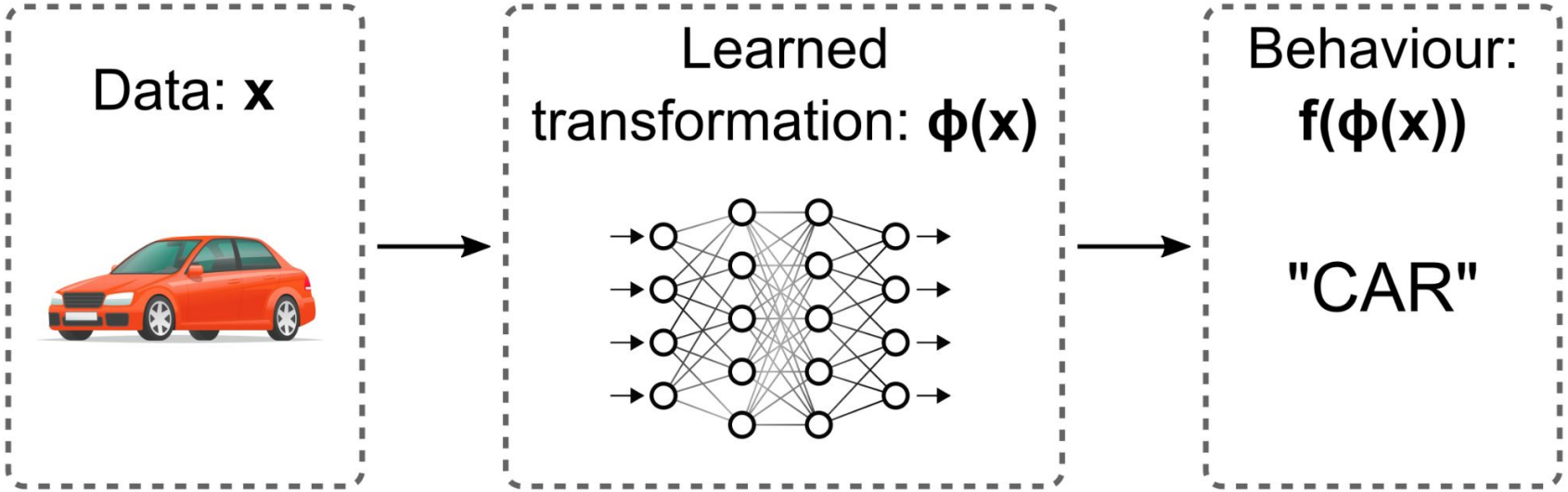
And on the 7th day God rested...
A job well done, no?

One answer: **because good representations are very helpful for behaving!**

Shallow learning avoids learning representations



Deep learning is the art of learning representations



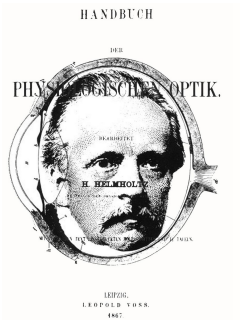
Can we learn useful representations without labels?

Labels are hard to find but unlabelled data is plentiful!

The goal of unsupervised and/or self-supervised learning is to learn useful representations from unlabelled data

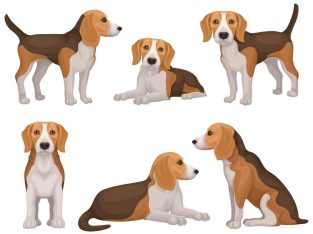
The key is to identify ***latent variables*** in the data

- Latent variables are variables that cannot be observed but which affect the variables you do observe
- Inferring latent variables is a major goal in statistics and machine learning, and some believe it is also the key to animal perception



Example: categories

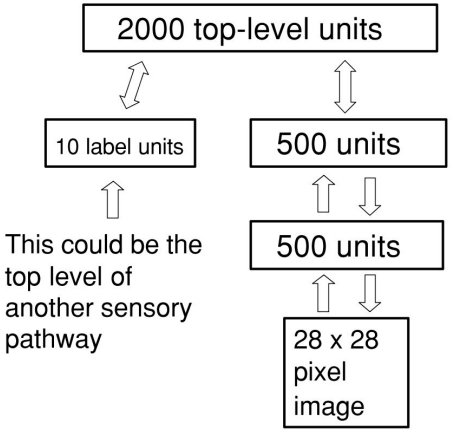
- You can’t directly observe the category of any object (objects don’t come with labelled tags usually)
- But, the category of an object will determine the variables you observe (e.g. the appearance of the object and the words people use for it)
- So, *it would be great if we could learn representations that capture latent variables like categories without labels*



This is a long-standing goal of deep learning

Many people associate deep learning with supervised learning based on labels

But the original goal of early deep learning models was unsupervised learning!



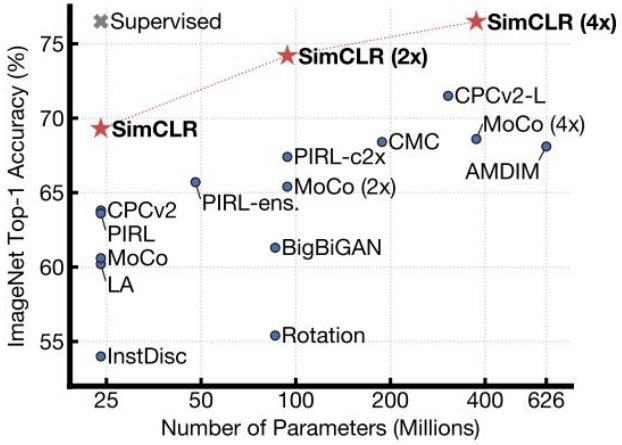
Example: In Hinton, Osindero & Teh (2006, *Neural Computation*, **18**:1527) they developed a means for unsupervised learning in deep-belief networks

The original promise is finally bearing fruit

Despite the early goal of unsupervised learning, deep learning in the 2010's became dominated by supervised learning because it worked surprisingly well and unsupervised learning wasn't working very well yet...

But recent advances have led to unsupervised (or self-supervised) learning methods that work really well.

The end of the dominance of supervised learning may finally be here!



https://ai.googleblog.com/2020/04/advancing-self-supervised-and-semi.html

A small note on terminology...

- The distinction between **unsupervised** versus **self-supervised** learning can be blurry sometimes
- Roughly it is this:
  - **Unsupervised learning** attempts to learn representations without labels by *not using any targets of any sort during training*, e.g. by using correlations in activity between units
  - **Self-supervised learning** attempts to learn representations without labels by *using the data itself to generate targets*, e.g. generating targets using the next word in a sentence
  - Put another way, self-supervised learning looks a lot like supervised learning in code, but there is a big difference related to the following question: *do you as a machine learning researcher have to actually ask someone to label the data or not?*
- Today we're really going to be mostly looking at self-supervised learning, because is what has worked the best to-date

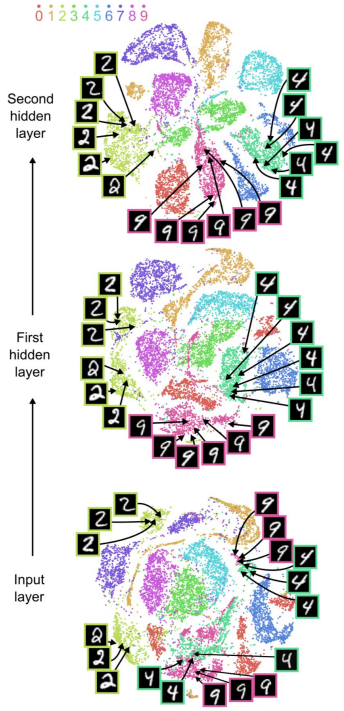
Lecture 1

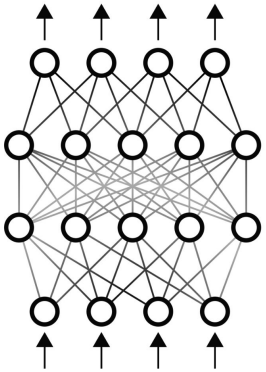
Why do representations matter?

- **Goal:** Convince yourself that representations do matter
- **Exercise:** Try doing classification with raw images versus pre-trained supervised representations

Supervised deep learning induces progressively more useful representations

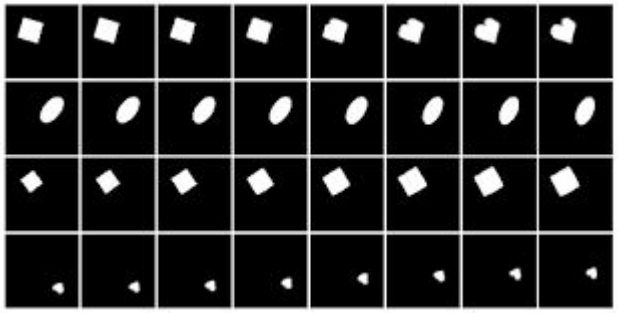
When we do supervised training of a neural network on some task, e.g. image categorization, the advantage of deep learning is precisely that it allows us to learn representations in the hidden layers that make the task easier at the final output layer





Exercise: demonstrate that representations matter

You must train a linear decoder to categorize shapes from the dSprites dataset



https://github.com/deepmind/dsprites-dataset

You will train off of:

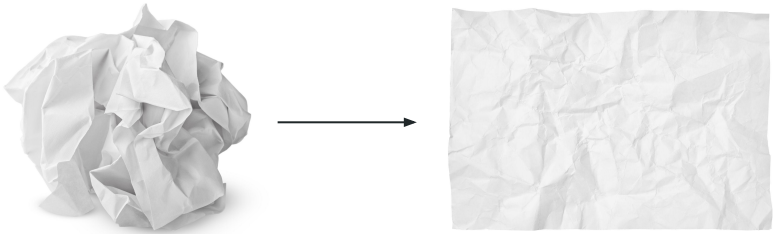
1. The raw images
2. The representations from the penultimate layer of a pre-trained (supervised) deep neural network

Answer this question: What leads to better accuracy?

Exercise: demonstrate that representations matter

Why use a linear decoder? What does that show?

- If you have good representations (e.g. your representations group key latent variables like categories together in the data) then these variables should be easy to decode from the representations using only linear projections



The pre-trained network you will use

- In this exercise we will give you a pre-trained convolutional neural network, and you will use the representations in its penultimate layer (before the final categorization happens)
- The network was trained to classify the three shapes in dSprites:
  - square, ellipse, heart
- It was trained on all the variations of position, scale, and orientation
- You must determine the accuracy that you can achieve by:
  - Training a linear decoder off of the dSprites images themselves
  - Training a linear decoder that is given the these convnet's representations
- Compare the accuracy you can get and think back to the original question in the intro: *Why didn't evolution connect your sensors directly to your muscles?*

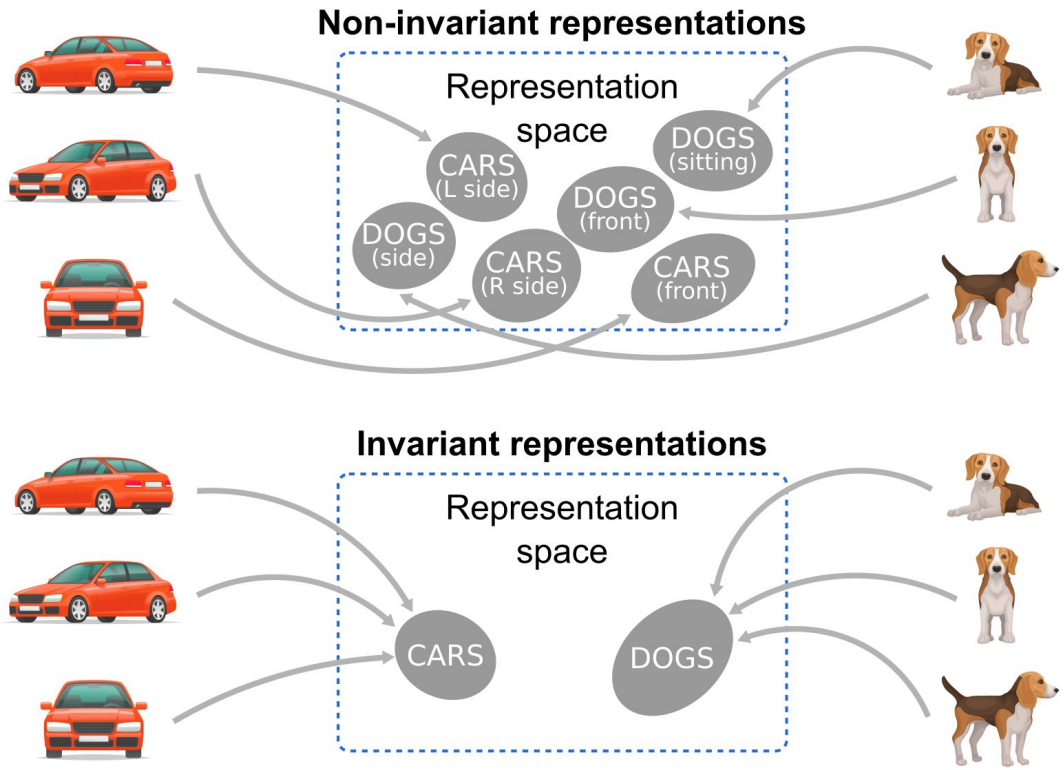
Lecture 2

Supervised learning and invariance

- **Goal:** Understand that supervised learning induces invariant representations
- **Exercise:** Perform representational similarity analysis of the representations in a supervised convnet, compare to the images themselves

What are the key properties of the representations induced by supervised learning? **Invariance!**

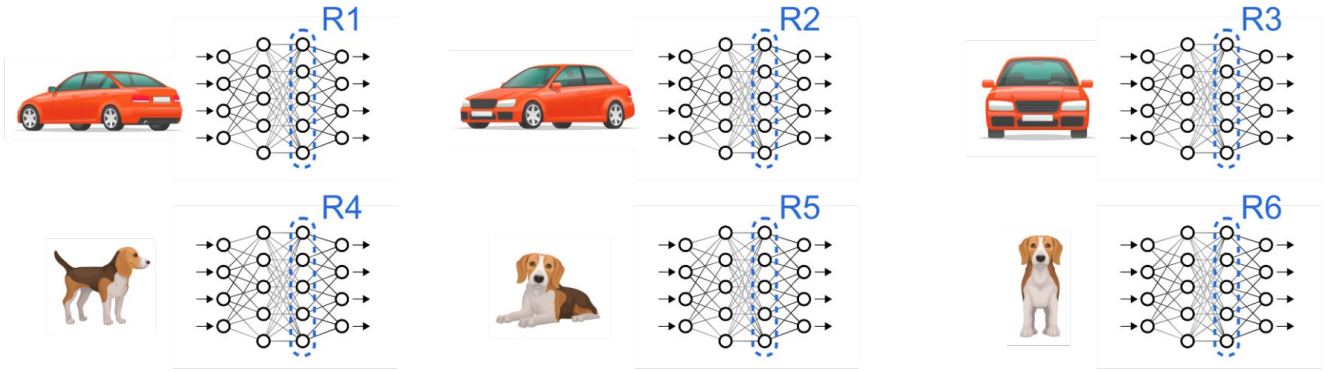
- One of the key properties of the emergent representations from supervised learning is **invariance**
- This means that you get a similar representation for the same object (or object class) regardless of the specific viewpoint, lighting, placement in the image, etc.
- Invariant representations make it much easier for a downstream circuit to recognize a given object regardless of the conditions under which it is viewed

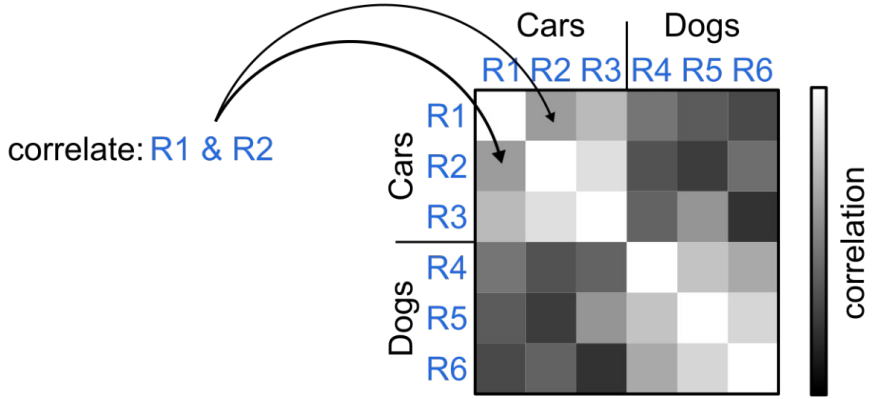

Non-invariant representations
Representation space
CARS (L side)
DOGS (sitting)
DOGS (front)
DOGS (side)
CARS (R side)
CARS (front)
Invariant representations
Representation space
CARS
DOGS

Exercise: verify supervised learning induces invariance

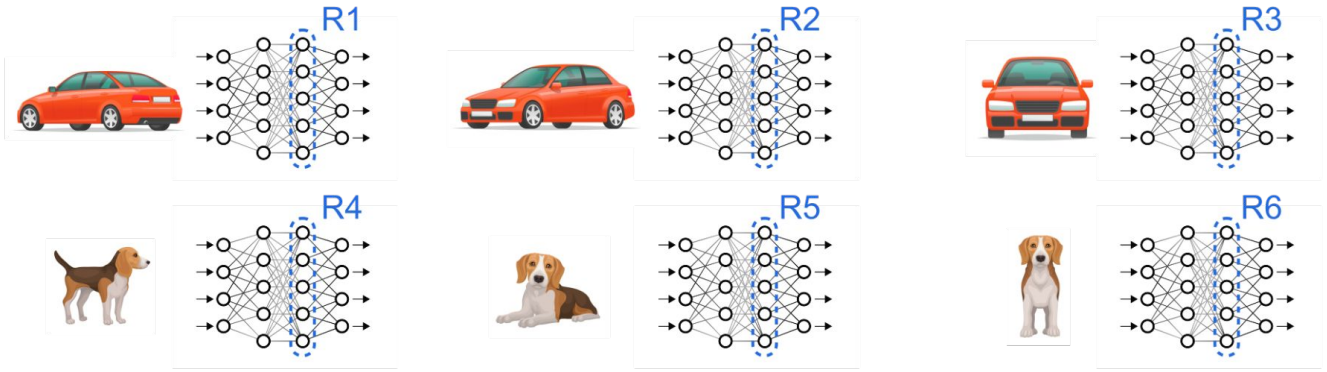
You must conduct a **Representational Similarity Analysis (RSA)** of a pre-trained, supervised, convolutional neural network (trained on dSprites)

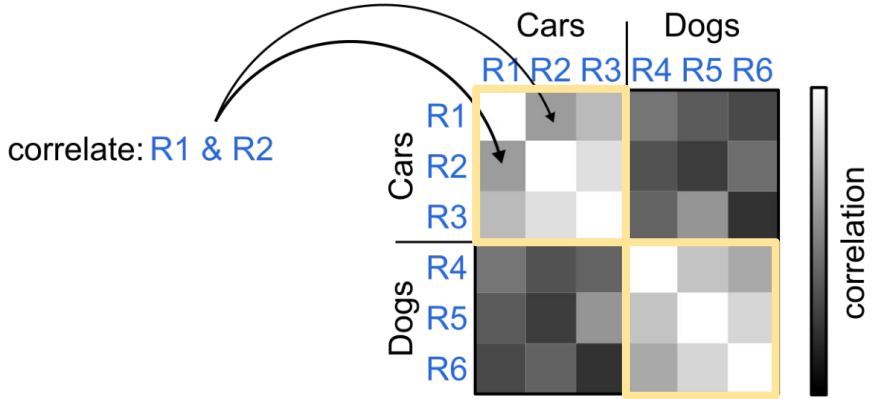
- RSA is a way of determining how similar the representations of different stimuli are to each other
- If a network has invariant stimuli, then RSA should show that objects from the same category are represented in a similar manner regardless of how they are transformed (e.g. orientation, location, etc.)





This is a **representational similarity matrix**





Invariant representations should have *higher values* for the same types of objects

Lecture 3

Random representations are not enough for good learning

Goal: Understand that random projections of the data can help, but do not suffice for effective learning

Exercise: Conduct RSAs and logistic regression using representations from a trained network and a random network

Some neural circuits use random connectivity

- For example, olfactory inputs into the mushroom body of fruit flies have a random structure to them (see Caron et al., 2013, *Nature* 497: 113-117)
- There is a large body of work showing that random projections can be useful for coding information
- Maybe we don't need to actually learn representations then? Maybe we can just use random connectivity and learn off of that?

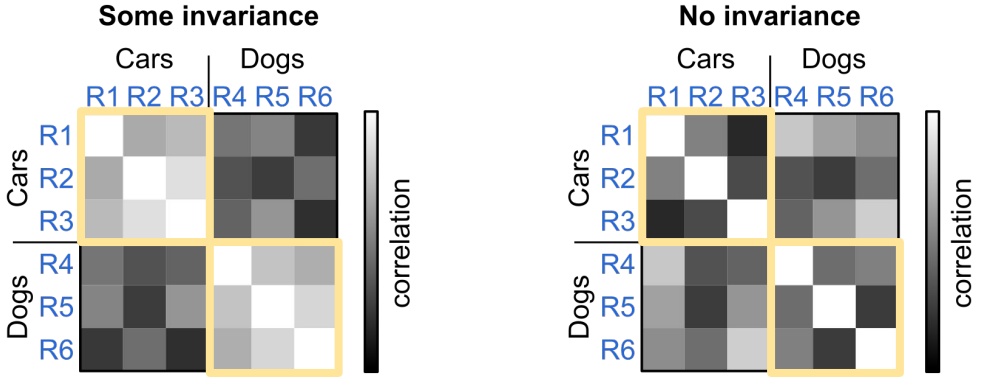
Exercise: test random projections

You must conduct RSA and logistic regression on dSprites using:

- A pre-trained convolutional neural network, trained in a supervised manner
- A convolutional neural network with random weights

Exercise: test random projections

- What do the RSAs look like? What level of accuracy can linear regression achieve?



- *What does this confirm about deep supervised training?*

Lecture 4

Generative models do not achieve what we need

Goal: Understand why generative approaches do not give us invariant representations

Exercise: Conduct RSAs and logistic regression using representations from a variational autoencoder

The original dream...

- Researchers realized long ago that we need to learn useful representations, and that we would ideally do so without labels
- The earliest ideas for achieving this were based on **generative models**
- The core idea is this:
  - If we can train a neural network to generate data from the some distribution, then surely that neural network will be using representations that capture the latent variables in the world that are ultimately responsible for that distribution
  - Example: if you have a network that can generate images of dogs and cars, and not weird half-dog/half-car images, then it must have representations that distinguish “dogs” vs. “cars”
  - Early models included Boltzmann machines and Helmholtz machines
  - The most recent incarnation of the same idea is variational autoencoders (VAEs)
    - See Kingma & Welling (2013): https://arxiv.org/abs/1312.6114

The disappointment...

- Despite getting really good at training generative models, the underlying representations learned by VAEs and other generative approaches didn't seem to help with downstream tasks (like classification) all that much
- Why? One issue is that good reconstruction requires representations that capture all details in the data, including details that are irrelevant to downstream tasks

Exercise: test representations learned from a VAE

You must conduct RSA and logistic regression on dSprites using:

- A VAE pre-trained to generate data from dSprites
- Examine the images reconstructed by the VAE, and the downstream performance of linear regression on the VAE representations of the images
- **Ask yourself:** *Does good image reconstruction seem to help with downstream classification?*

Lecture 5

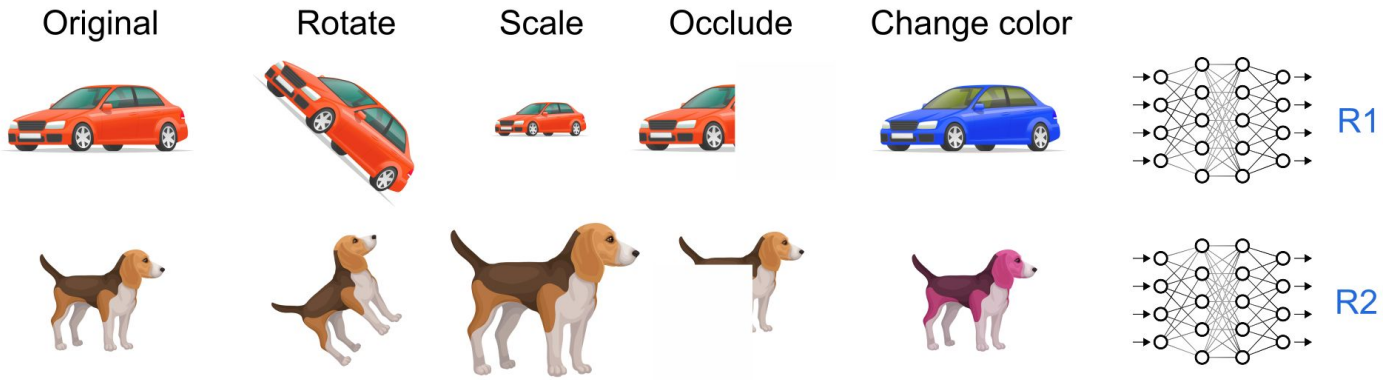
The modern approach in self-supervised learning

Goal: Understand use of transformations for training invariant representations

Exercise: Examine potential data transformations used for training invariant representations

What if we aim for invariance to transformations?

- If our goal is invariant representations, then maybe we could just train directly towards that goal using data transformations/augmentations?



Exercise: explore transformations on dSprites

Try out a few potential affine transformations on dSprites

- Rotation, scaling, translation
- **Ask yourself:** *What combinations of affine transformations would be important for learning useful invariances?*

Lecture 6

How to train for invariance with data transformations

Goal: Understand the modern approach to self-supervised learning, using transformations

Exercise: Train a SimCLR network on dSprites, then categorize with the learned representations

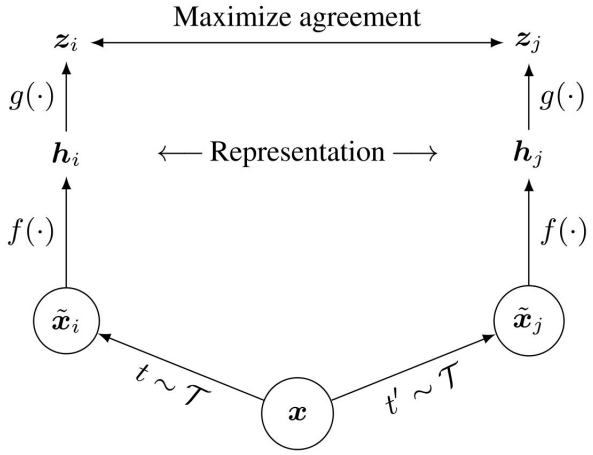
A Simple Framework for Contrastive Learning (SimCLR)

- In a paper published in 2020 Chen et al. demonstrated a simple approach for learning invariant representations
- Their approach (SimCLR) uses this strategy:
  - Take an image
  - Apply two different transformations to it
  - Run the two images through a convnet to get a representation
  - Take the representation and run it through a multi-layer perceptron
  - Train the final output to be the as similar as possible for augmentations of the same image (***positive samples***), but as different as possible for different images (***negative samples***, more on this in Lecture 8)

https://ai.googleblog.com/2020/04/advancing-self-supervised-and-semi.html

A Simple Framework for Contrastive Learning (SimCLR)

SimCLR uses a ***contrastive loss*** to maximize agreement between positive samples while minimizing agreement between negative samples

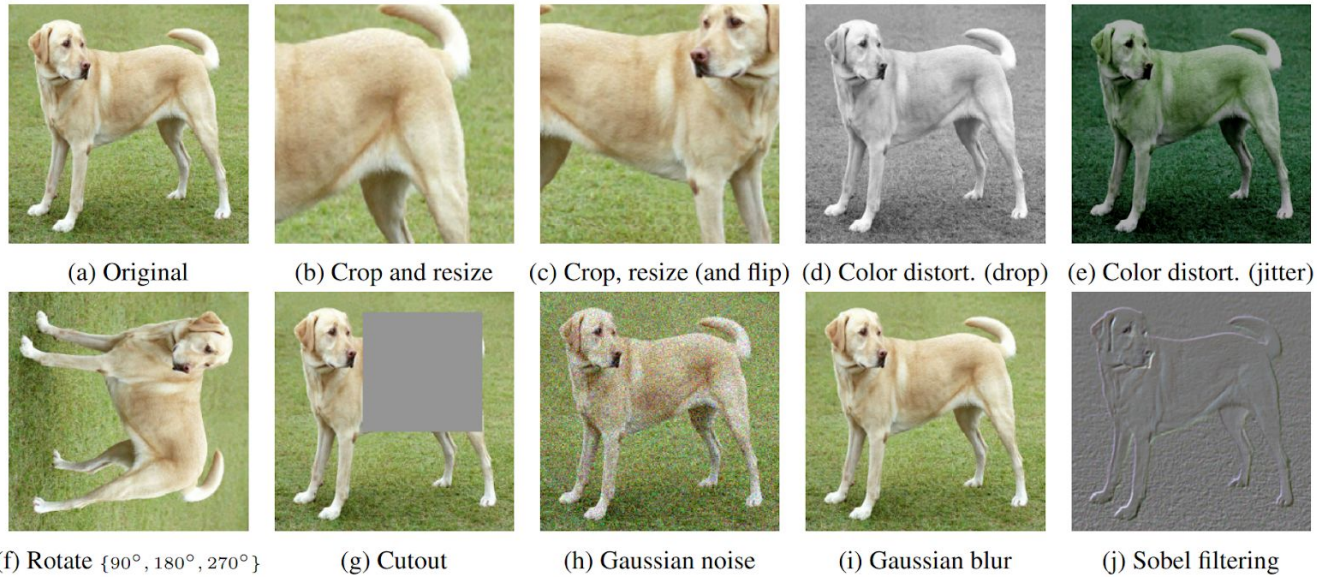


$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

$$\text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v} / \|\boldsymbol{u}\| \|\boldsymbol{v}\|$$

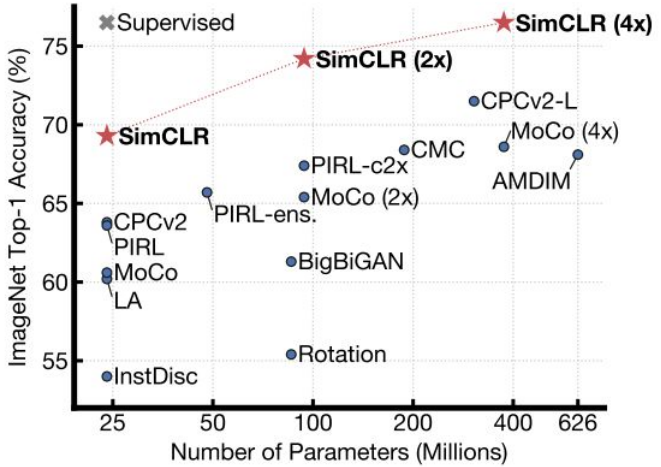
A Simple Framework for Contrastive Learning (SimCLR)

- The transformations/augmentations used are critical for achieving good results



A Simple Framework for Contrastive Learning (SimCLR)

- After training with no labels, the representations that are output by the convnets can then be used for downstream tasks, like categorization
- Pre-training with SimCLR allows one to achieve the same level of accuracy as deep supervised learning with a linear classifier!



Exercise: test representations learned by SimCLR

You must conduct logistic regression on dSprites using the representations of a pre-trained SimCLR network

- **You must try to fill in the code for calculating the numerator and denominator in the loss function** (but note, you will not have time to train the network fully, so you can use the pre-trained network for the regression)
- Compare regression on the pre-trained network to the other tests you've done
- **Ask yourself:** *How does the accuracy compare to the other methods you've tried so far (straight from images, random, VAEs, etc.)?*

Lecture 10

Un/self-supervised learning and biased datasets

Goal: To understand the potential impact of biased datasets on representation learning, and how design choices impact this

Exercise: Examine the accuracy of the models when trained on a biased dataset

Many datasets are biased...

By this we mean: Datasets can contain correlations and restrictions that do not reflect the actual range of scenarios that we want the model to perform well on

What happens when we conduct unsupervised or self-supervised learning on biased datasets?

That will depend on your model design and choices!

- If you have a model that is designed to reflect the statistics of the dataset (e.g. a generative model), it will necessarily reflect those biases
- If it is designed to be invariant to a range of different alterations of the data (including conditions that don't exist in the training data) then it may not...

Exercise: Train on a biased dataset, test on unbiased

You will train the logistic regression off of representations from a supervised network, a VAE, a random network, and a SimCLR network. These networks will be pre-trained with a biased version of dSprites where:

- Hearts only ever appear on the left
- Ovals only ever appear in the centre
- Squares only ever appear on the right
- **Ask yourself:** *What happens to accuracy on an unbiased test set when the model is trained on the biased dataset? Are there differences between the models? What would you predict on a real-world biased dataset?*

Conclusions

Back to the original dream...

As noted at the start of today, the idea that neural networks should be able to learn representations of latent variables without human labelled data has been around since their inception, and early deep learning was actually more concerned with unsupervised learning

Focusing on what matters: invariance

- But, for a long time we never actually saw the development of unsupervised learning techniques that learned representations that actually helped with downstream tasks
- This is why supervised learning ended up becoming so dominant in the field for so many years
- The thing that really helped: *focussing less on data generation and more on invariant representations*

But there are many different types of invariance!!!

- Depending on the design choices in your model (e.g. transformations or loss function), you end up emphasizing different forms of invariance
- **Example**: SimCLR emphasizes invariance to transformations, but promotes variance between different images
- But, what if we were to train on videos and use the future sensory inputs as our target for self-supervision?

Prediction encourages two types of invariance

Movement invariant prediction

"I will see an orange car"

Object invariant prediction

"There will be leftward movement"

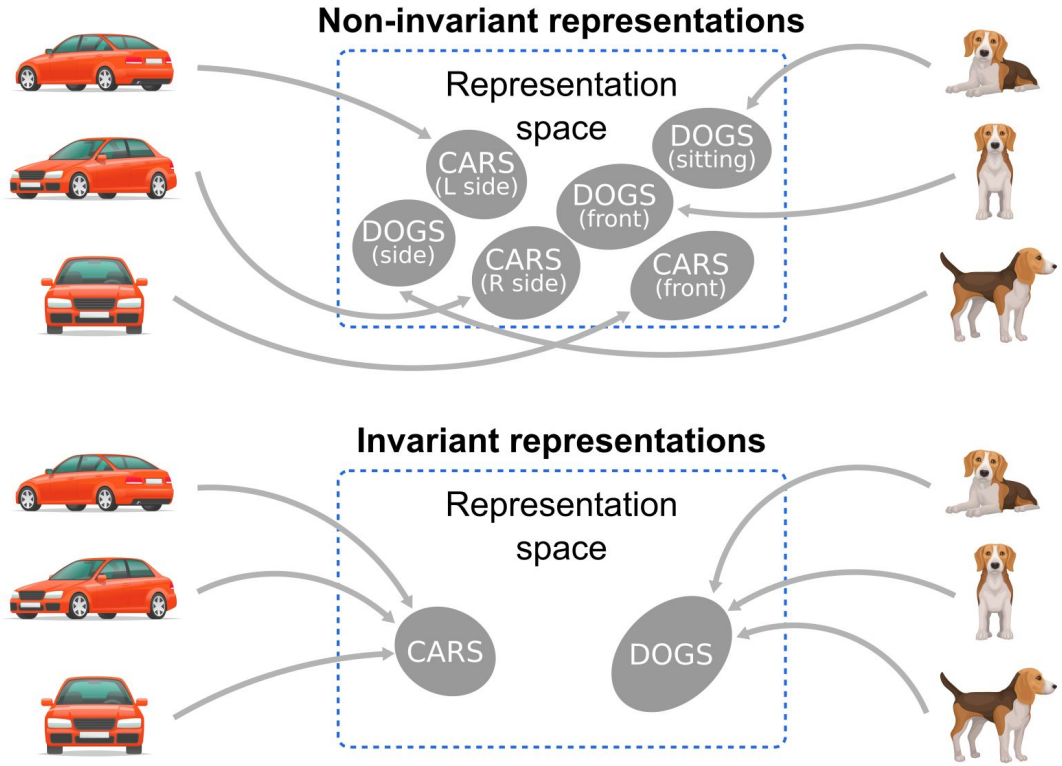
We are not passive... what about actions?

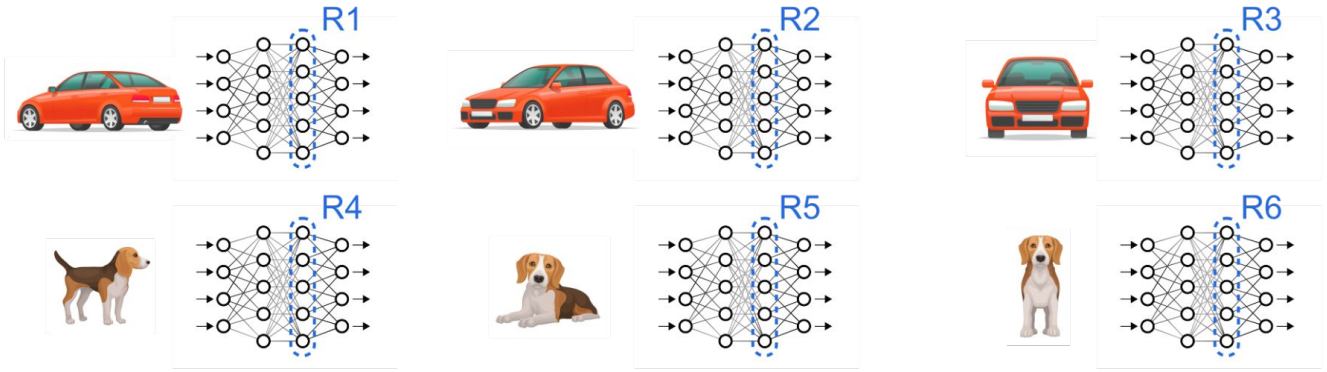
Lecture 7

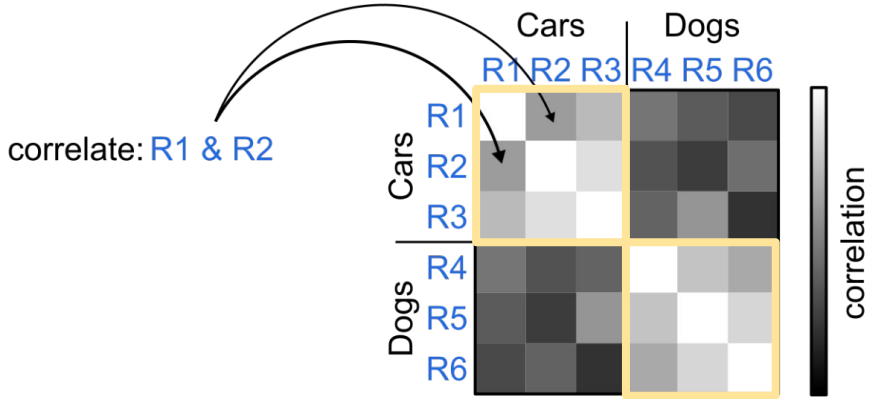
Self-supervised learning induces invariant representations

Goal: Convince yourself that self-supervised learning achieves invariant representations similar to those achieved with supervised training

Exercise: RSA on SimCLR representations


Non-invariant representations
Representation space
CARS (L side)
DOGS (sitting)
DOGS (front)
DOGS (side)
CARS (R side)
CARS (front)
Invariant representations
Representation space
CARS
DOGS





Invariant representations should have *higher values* for the same types of objects

Exercise: RSA on representations learned by SimCLR

Perform the same RSA on the representations of dSprites learned by the SimCLR network

- Compare to supervised representations and representations from a VAE
- **Ask yourself:** *Which looks more invariant, the VAE representations or the SimCLR representations?*

Lecture 8

Avoiding representational collapse

Goal: Understand why negative samples are critical for SimCLR to avoid representational collapse

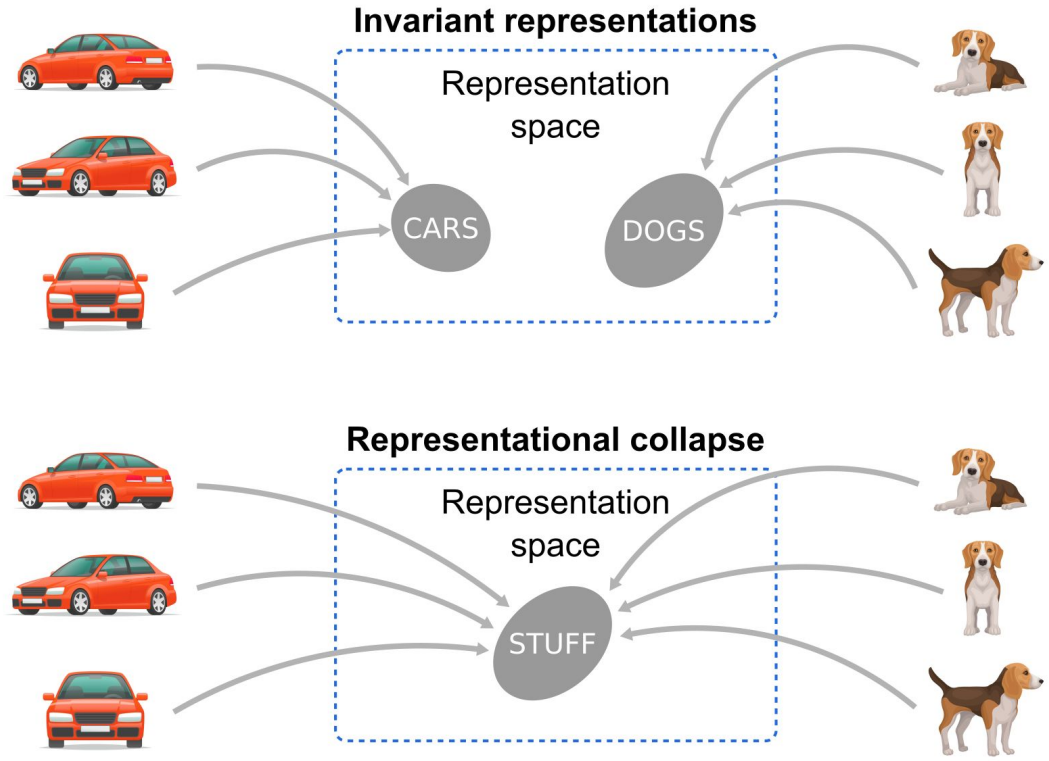
Exercise: Conduct RSA on SimCLR network trained with fewer negative samples

SimCLR needs negative samples

- As already noted, SimCLR both trains the network to both:
  - Match the representations for augmentations of the same image (these are called **positive samples**)
  - Differentiate the representations for augmentations of different images (these are called **negative samples**)
- But, if the goal is invariant representations, why do we need the negative samples?

Avoiding **representational collapse**

- If you only train a network to match representations for invariance, and never to differentiate, then the network can find a trivial, crappy solution:
  - Represent everything with the same representation!!!!
  - This is known as **representational collapse**
  - The loss function one chooses is critical to avoid representational collapse!


Invariant representations
Representation space
CARS
DOGS
Representational collapse
Representation space
STUFF

Exercise: RSA on representations learned by SimCLR with few negative samples

Perform the same RSA on the representations of dSprites learned by the SimCLR network, but this time use only a couple of negative samples during training

- Compare to regular SimCLR representations
- Examine the histograms of the representational similarity matrix values
- **Ask yourself:** *What happens to the representations without sufficient negative samples? Are they more or less similar across categories? How does this affect downstream categorization?*

Lecture 9

Few-shot supervised learning with self-supervised learning

Goal: Understand that self-supervised pre-training can reduce the need for labels

Exercise: Examine the accuracy achieved by logistic regression from our different representations using varying numbers of labels

Using self-supervised learning to reduce label dependence

- Self-supervised learning techniques can be applied to a dataset without labels
- Afterwards, if a few labels are available, supervised learning can be applied, and the same level of accuracy can be achieved with far fewer labels
- Example: Training a linear decoder off of SimCLR representations gets the same level of accuracy on ImageNet as can be achieved with fully supervised training using **one tenth of the labels**

Method	Architecture	Label fraction (Top 5) 1%	10%
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(∗)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Table 7. ImageNet accuracy of models trained with few labels.

Why does this work?

If you already have good, invariant representations of your data then the classification task is a simple optimization procedure

- The self-supervised learning has already done the hard part of deep learning, i.e. the representation learning
- This is what makes deep learning data hungry
- So, if you can do representation learning well without labels, then your supervised learning with labels will not actually be data hungry

Exercise: Regression with varying numbers of labels

Perform logistic regression on dSprites using representations formed by the various approaches we've discussed today

- Examine what happens when you provide the regression with a decreasing number of labels
- **Ask yourself:** *Which approach allows you to achieve decent classification when labels are limited?*
    - *Are there differences between training and testing data? Does this differ for the different representations?*