BIO-INSPIRED COMPUTING: APPLICATIONS AND INTERFACES

REPRESENTATION LEARNING

Slides created by Jo Plested

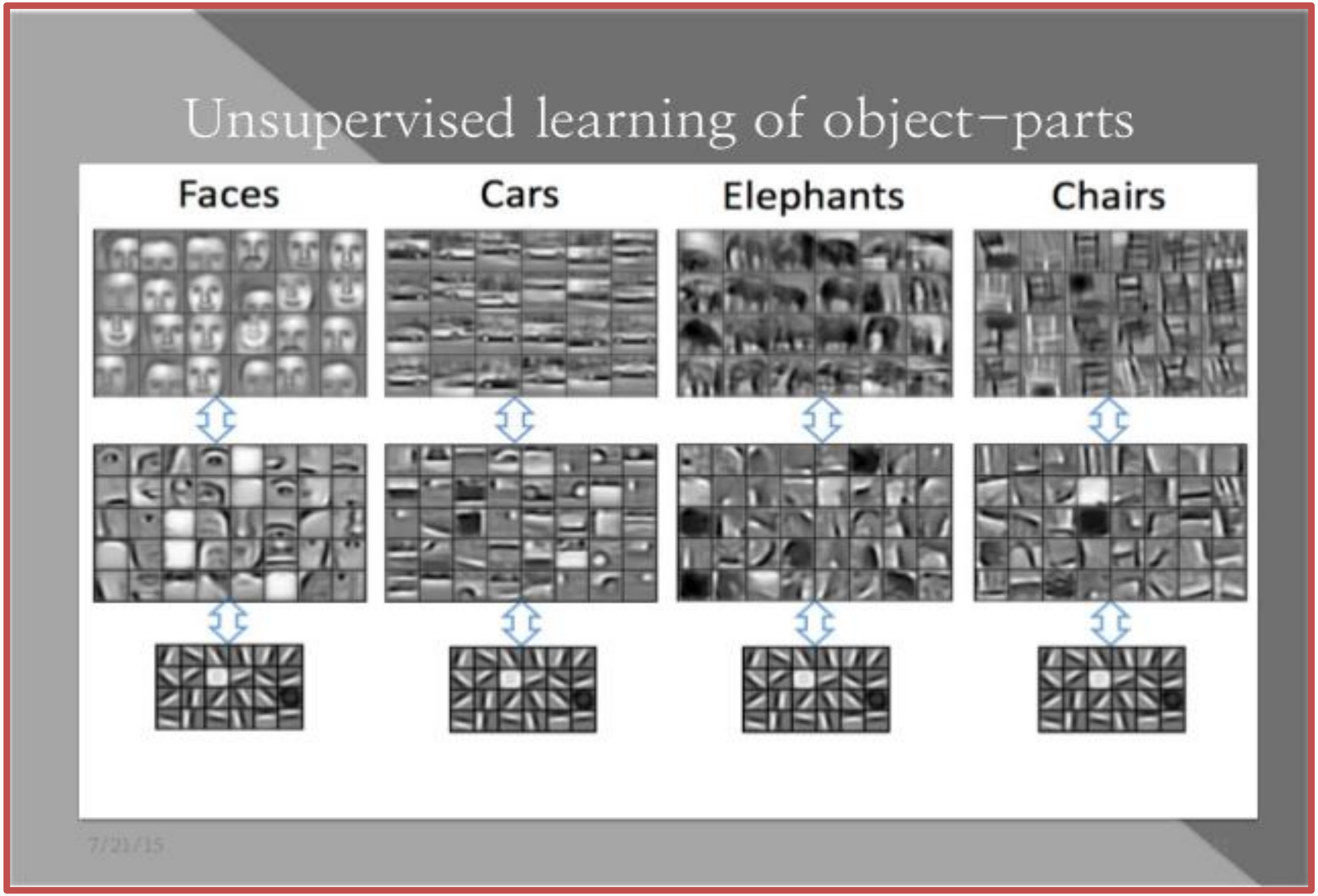
LECTURE OUTLINE

- Representation Learning
- Transfer Learning
- Unsupervised (self-supervised) Learning
- Semi-supervised Learning

REPRESENTATION LEARNING

- Deep learning can be thought of as representation learning
- The layers learn the best representation of the data to perform the desired task rather than a domain expert hand designing the representation
- One of the top conferences in deep learning is ICLR – the International Conference on Learning Representations

REPRESENTATION LEARNING



citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.149.802&rep=rep1&type=pdf

TRANSFER LEARNING

- Deep learning techniques have been proven to be state of the art in learning features from image, video, audio and text data, but they are dependent on having large amounts of training data
- Using deep learning with small datasets usually results in overfitting

TRANSFER LEARNING

- In many domains it is difficult or prohibitively expensive to get large amounts of labelled training data, but unlabelled or related data is available
- The hardware used to train very Deep Neural Networks (DNNs) is also often prohibitively expensive. Top research facilities like Google and Facebook train their largest models for weeks on 100's of GPUs
- For these reasons it's common for very large DNNs not to be trained entirely from scratch

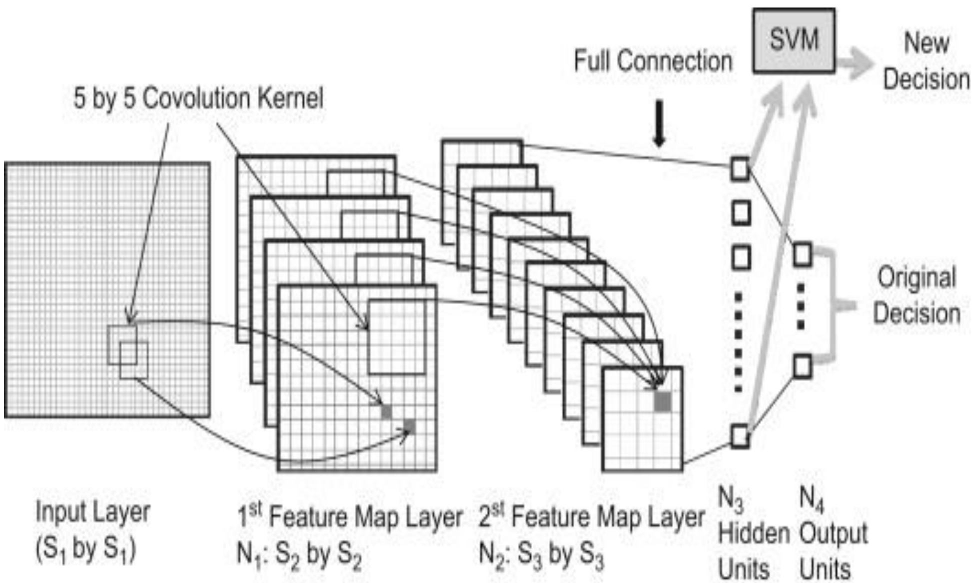
TRANSFER LEARNING

- Transfer learning is where we train the weights of our model on a related (source) dataset and transfer those pre-trained weights over to the model for our target dataset rather than initialising the weights from scratch.
- Transfer learning can be used to reduce both the amount of labelled data and time needed for training a DNN
- There are a number of different ways that transfer learning can be performed depending on the target task and the data available

TRANSFER LEARNING - METHODS

Using a fully trained DNN as a fixed feature extractor

- best if new dataset is very small and similar to original



Example: take a CNN pretrained on ImageNet (1.2 million labelled images), remove the final layer and treat the previous layer as a feature extractor. Train a linear classifier (eg. SVM or Softmax) on the new dataset using the extracted features as input

Niu, Xiao-Xiao, and Ching Y. Suen. "A novel hybrid CNN–SVM classifier for recognizing handwritten digits." Pattern Recognition 45.4 (2012): 1318-1325.

TRANSFER LEARNING - METHODS

Fine tuning a fully trained DNN

- Best if target dataset is small to medium sized
- Transfer some layers of weights from a large DNN trained on a similar large dataset
- Fully train layers that are not transferred (reinitialised as random weights)
- Fine tune with small learning rate some layers of transferred weights

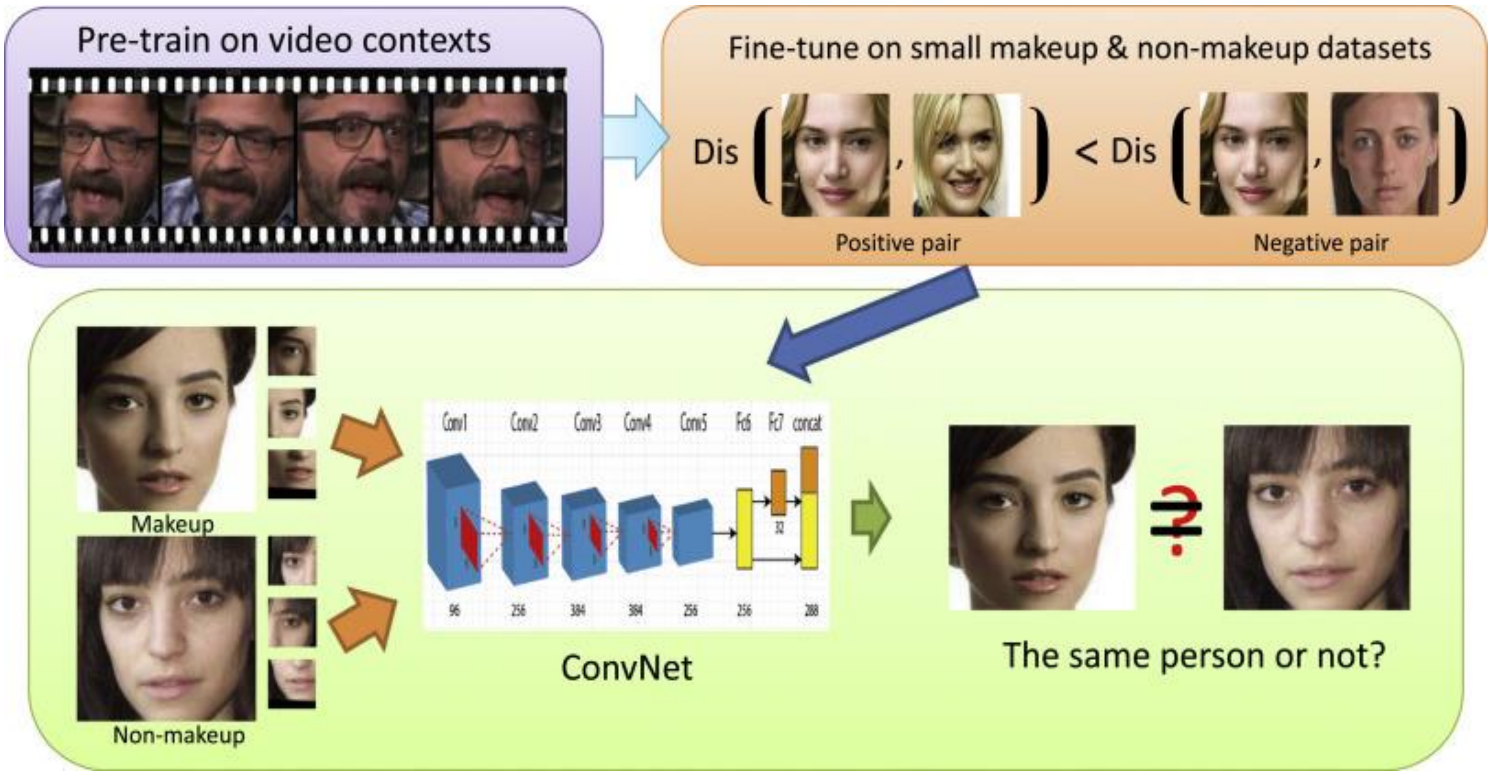
TRANSFER LEARNING - METHODS

Fine tuning a fully trained DNN

- Starting from the output layer the number of layers to be reinitialised and fully trained depends on the amount and similarity of the target dataset. As does the number of layers to be fine tuned

TRANSFER LEARNING - METHODS

Fine tuning a fully trained DNN



Liusi-group.com/pdf/
faceverification-pr-2017.pdf.

TRANSFER LEARNING - METHODS

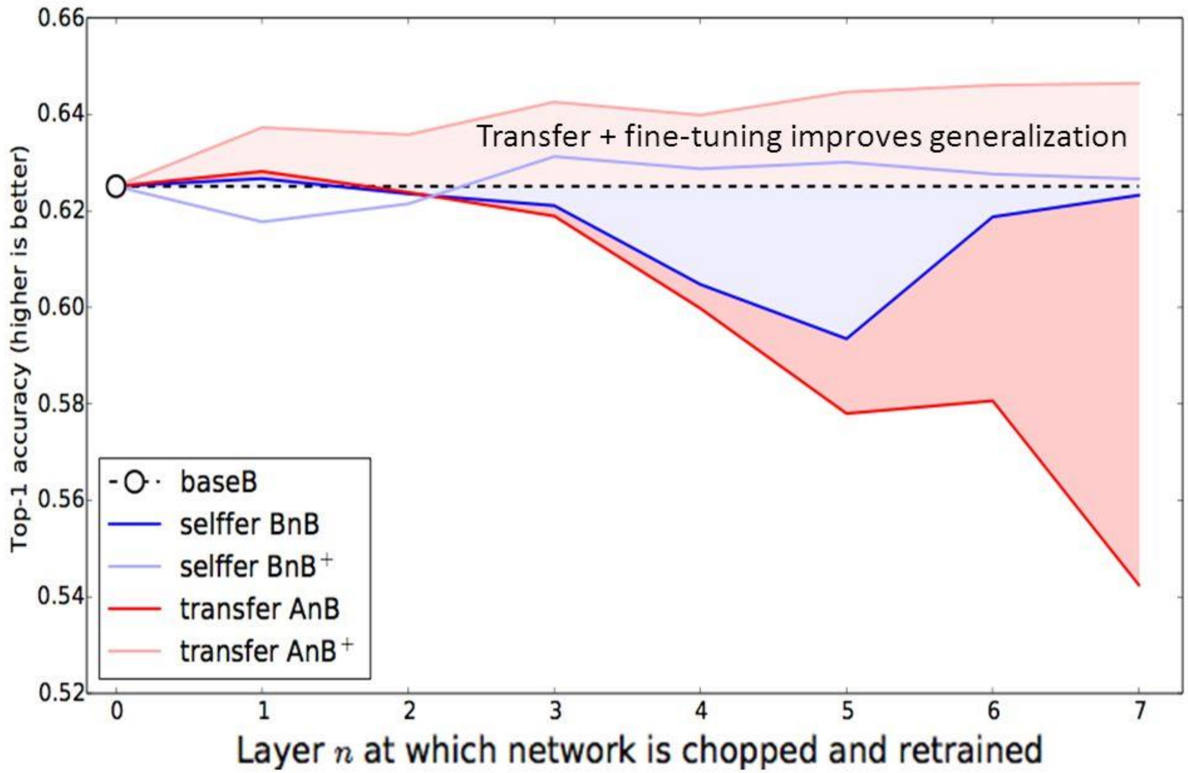
Fine tuning a fully trained DNN – how does it compare to no fine tuning?

Paper: papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf

- Lower level features in images 'appear not to be specific to a particular dataset or task, but general in that they are applicable to many datasets and tasks'
- 'the transferability of features decreases as the distance between the base task and target task increases, but transferring features even from distant tasks can be better than using random features'

TRANSFER LEARNING - METHODS

Fine tuning a fully trained DNN – how does it compare to no fine tuning?



slide credit Jason Yosinski

Transfer AnB: Pretraining on A final training and testing on B. Layers which are transferred are not retrained at all
\+ layers that are transferred are fine tuned.
Selffer BnB: Pretraining on B final training and testing on B for comparison

TRANSFER LEARNING - METHODS

Unsupervised pre-training

- Limited labelled data but large amounts of unlabelled data from the same or similar dataset
- Two motivations
  1. Learning about the input distribution can help with learning a mapping from input to output
  2. The choice of initial parameters for a DNN can have a significant regularising effect. Helps to reduce overfitting
- There are many different unsupervised DNN methods

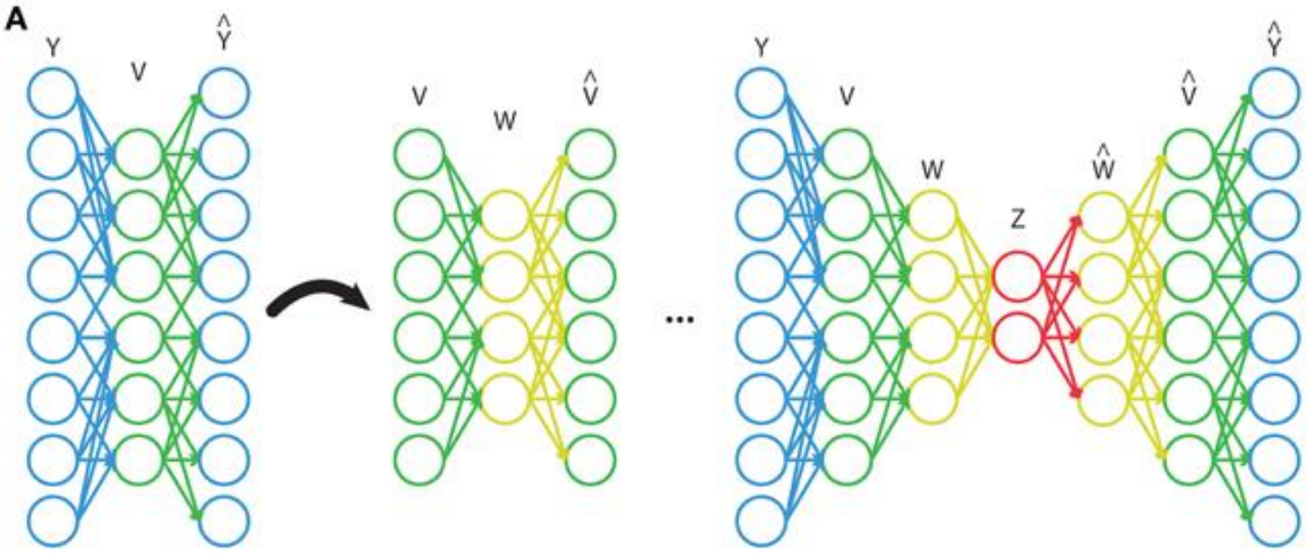
UNSUPERVISED LEARNING

- There are a number of general unsupervised learning methods and many more that are specific to the type of data being learned from
- Unsupervised learning also plays a big part in the history of DNNs and played a key role in the revival and current success of DNNs
- Before improvements like ReLU activation functions, batch normalisation, dropout and others it wasn't possible to train all the weights of a DNN together from scratch

GREEDY LAYERWISE UNSUPERVISED PRETRAINING

- Greedy layerwise training procedures with unsupervised criteria have a long history of being used to get around the difficulties of training a DNN from scratch
- Rather than being initialised randomly a DNN is built up one layer at a time with each layer learning a representation of the data with an algorithm such as an autoencoder and that layers output used as input to the next layer

GREEDY LAYERWISE UNSUPERVISED PRETRAINING

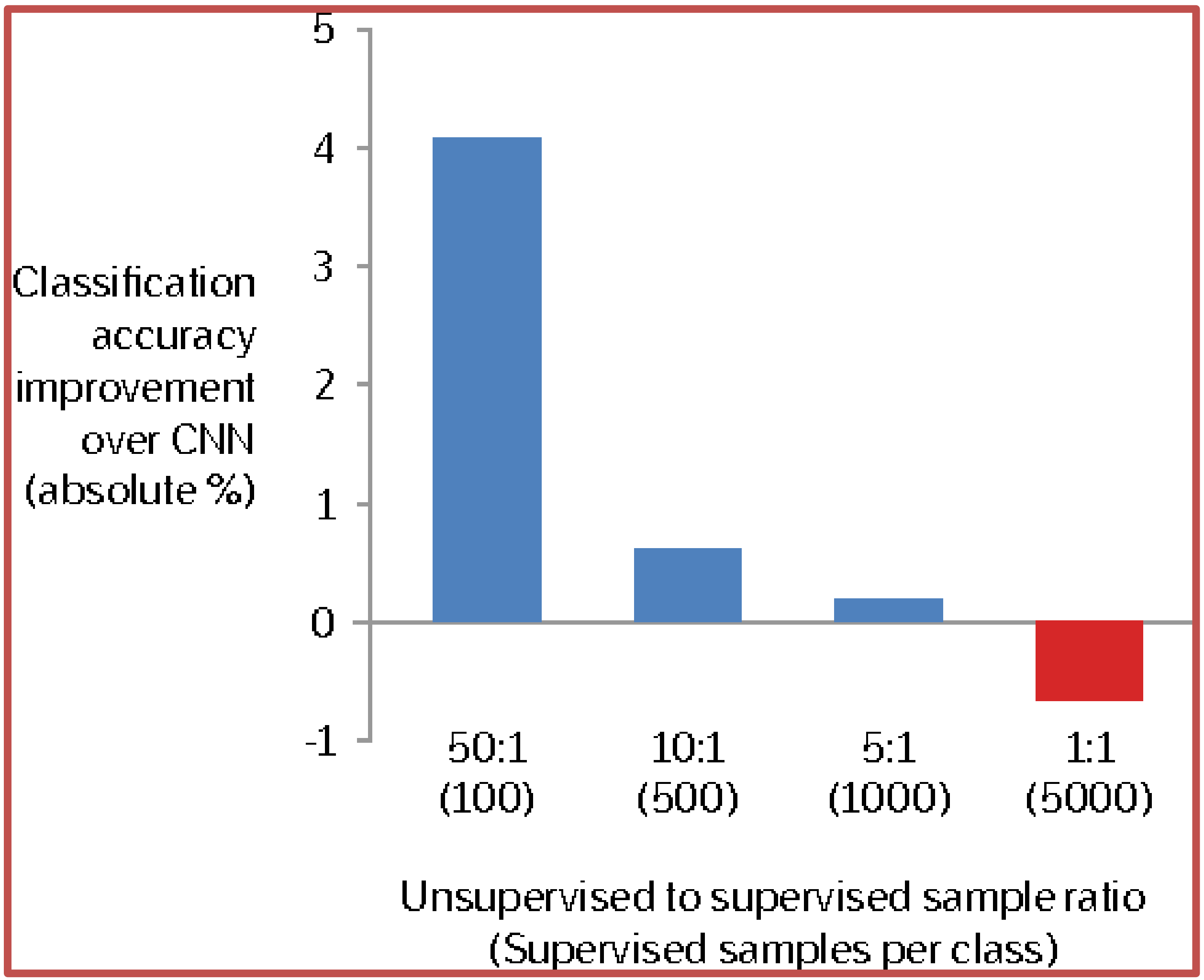


GREEDY LAYERWISE UNSUPERVISED PRETRAINING

- Greedy layerwise unsupervised pretraining is still useful even with modern techniques when the number of labelled samples is low, but can hurt performance when the number is high

Paper: https://arxiv.org/pdf/1412.6597

GREEDY LAYERWISE UNSUPERVISED PRETRAINING



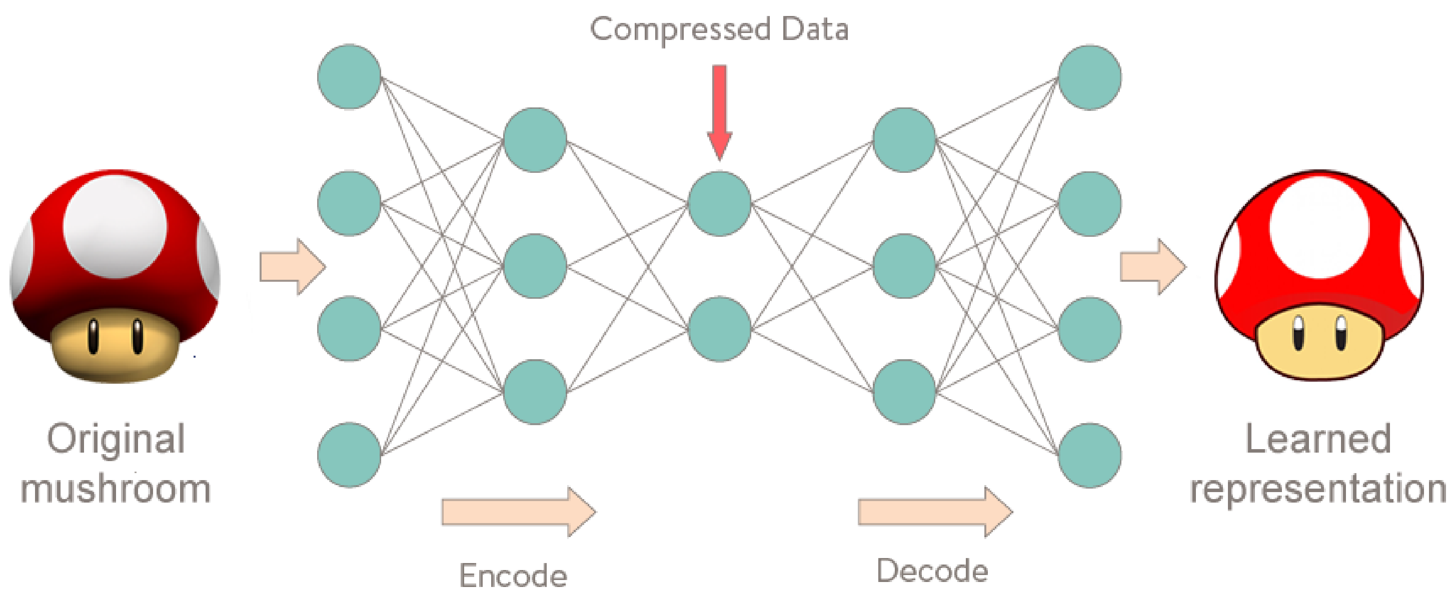
UNSUPERVISED LEARNING – AUTOENCODERS

- An autoencoder is a neural network that is designed to learn a representation of its input by attempting to copy its input to its output
- It can be thought of as two separate functions. An encoder and a decoder
- Autoencoders are generally restricted so they can't just learn to map input to output exactly and have to prioritise important features that are often useful for other tasks

UNSUPERVISED LEARNING – AUTOENCODERS

Ways of restricting autoencoders

1. Limit the dimension of h to be less than the input. In this case the cost function is just *L(x,g(f(x)))*, where x is the input, f is the encoder, g is the decoder and L is a loss function that penalises *g(f(x))* when it is dissimilar to x e.g. mean squared error

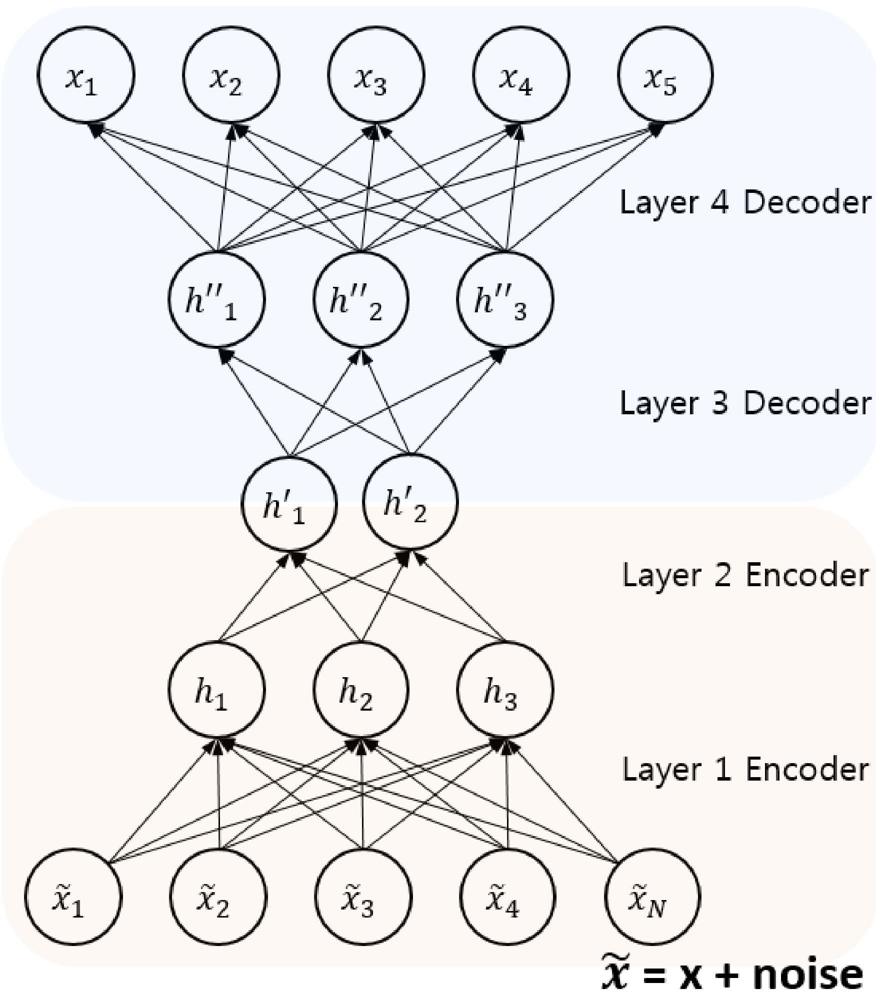


https://medium.com/@curiousily/credit-card-fraud-detection-using-autoencoders-in-keras-tensorflow-for-hackers-part-vii-20e0c85301bd

UNSUPERVISED LEARNING – AUTOENCODERS

2. Regularisation
   - encourage sparseness with a sparsity penalty: $L(x,g(f(x))) + \Omega(h)$
   - penalise derivatives. Forces the encoder to learn a representation that doesn't change much as the input changes. $L(x,g(f(x))) + \Omega(h,\Delta x)$, where $\Omega(h,\Delta x)$ is a term that penalises the derivates of x. Referred to as a contractive autoencoder (CAE)
   - Denoising autoencoders (DAE) minimise $L(x,g(f(\tilde{x})))$, where $\tilde{x}$ is a corrupted copy of x. So they are aiming to undo the corruption of x

UNSUPERVISED LEARNING – AUTOENCODERS



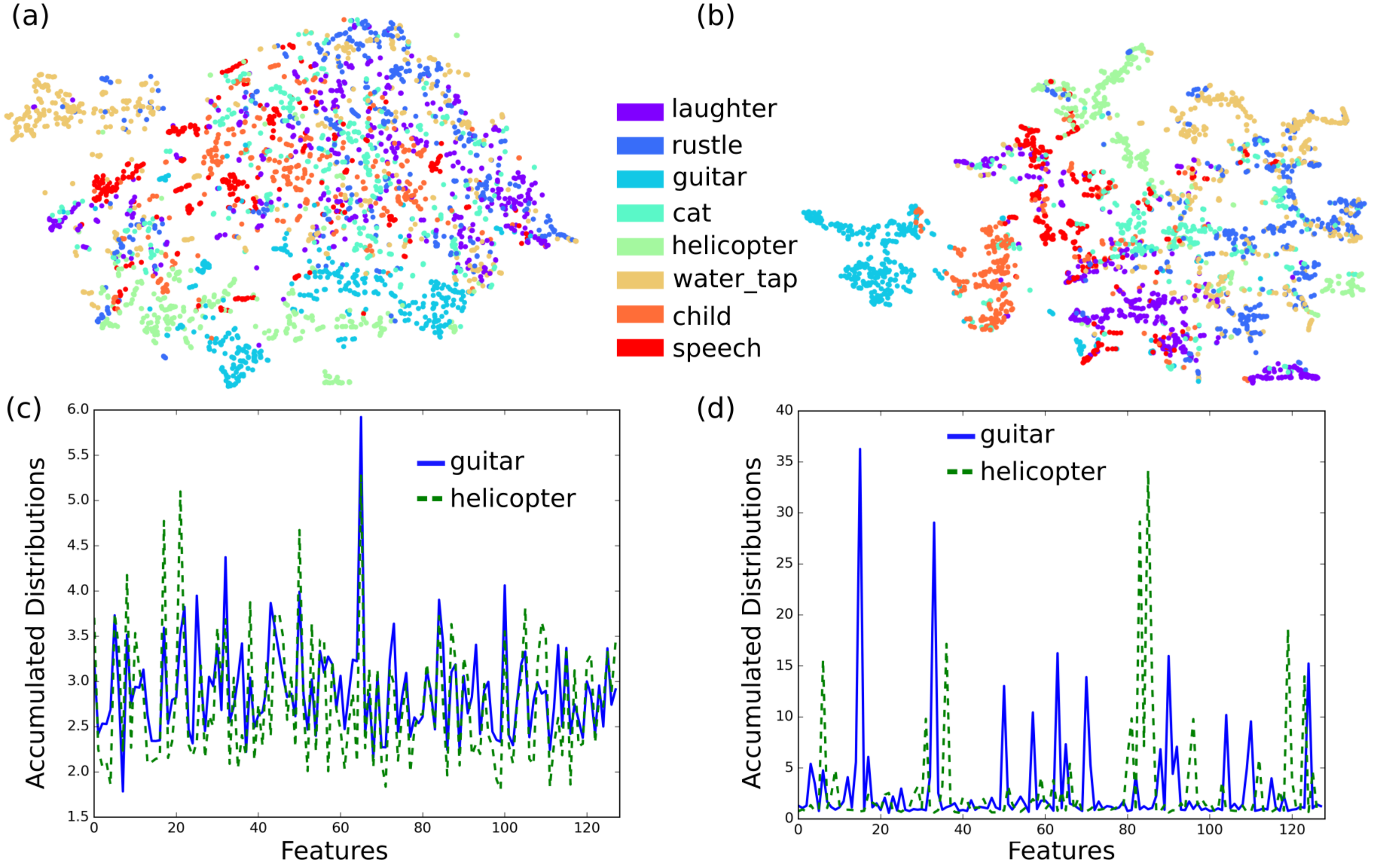
A denoising autoencoder: the input is the original data with noise added to corrupt it and the autoencoder attempts to recreate the original data as output

https://wikidocs.net/3413

UNSUPERVISED LEARNING – AUTOENCODERS

Unsupervised feature learning for audio analysis

Paper: https://arxiv.org/pdf/1712.03835



Training an autoencoder on audio events resulted in better clustering of like events in feature space than the baseline

UNSUPERVISED LEARNING – CONTRASTIVE LEARNING

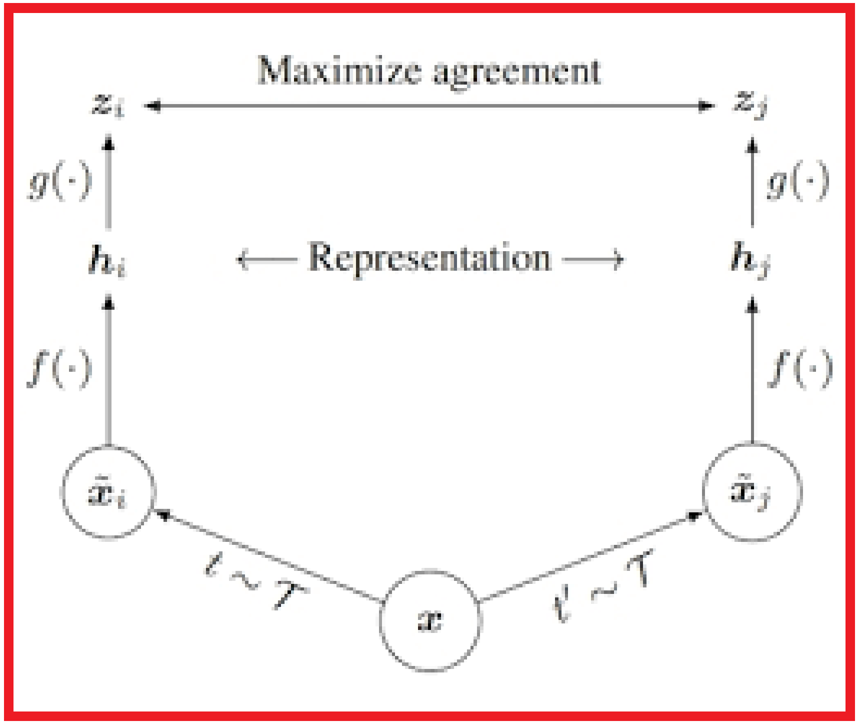
- Train model to predict whether or not two transformed *views* were created from the same input.
- Need to provide a list of transformation (families) to be randomly applied to inputs. These transformations should capture natural variance in the dataset. representations which are invariant to a specific set of transformations.
- Contrastive learning is most commonly used on image and audio data.

UNSUPERVISED LEARNING – CONTRASTIVE LEARNING

SimCLR

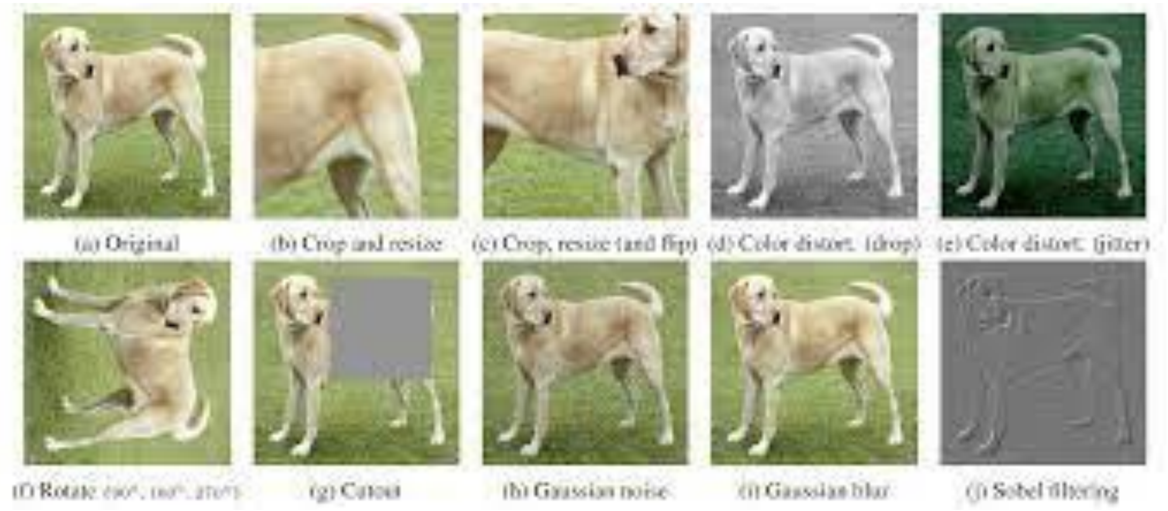
For each image in the batch, randomly sample 2 different sequences of transformations and apply them, to create two different *views*.

Train neural network such that output vectors of views from the same image are similar (in terms of angle), and vectors from different images are different.



UNSUPERVISED LEARNING – CONTRASTIVE LEARNING

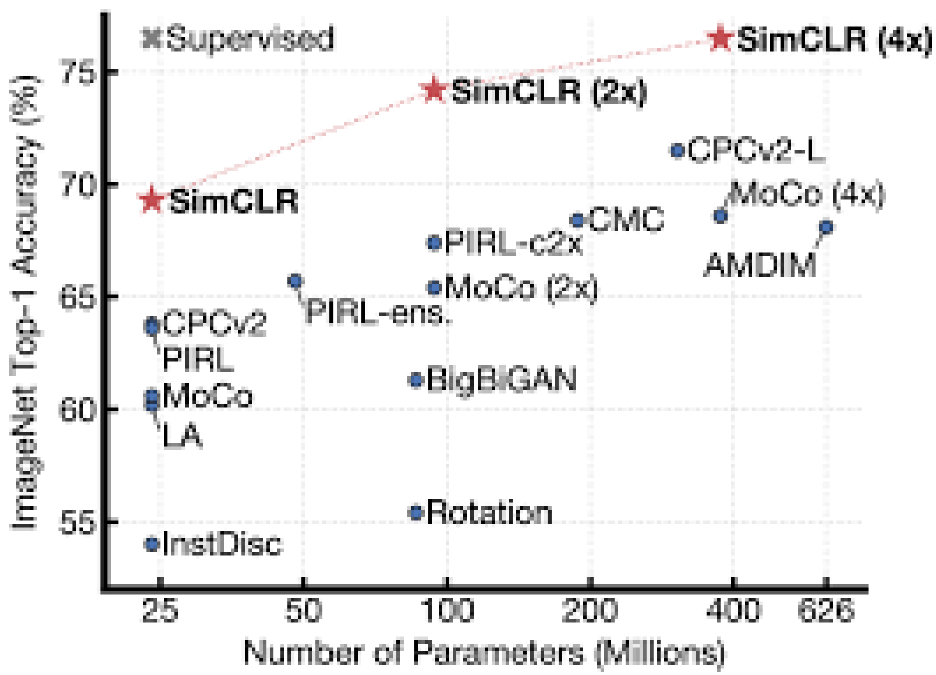
These are the transforms considered in the SimCLR paper. The final version only uses Crop, resize and flip, gaussian noise, and colour jitter.



UNSUPERVISED LEARNING – CONTRASTIVE LEARNING

SimCLR needs to use a much larger model and much larger batches compared to supervised learning.

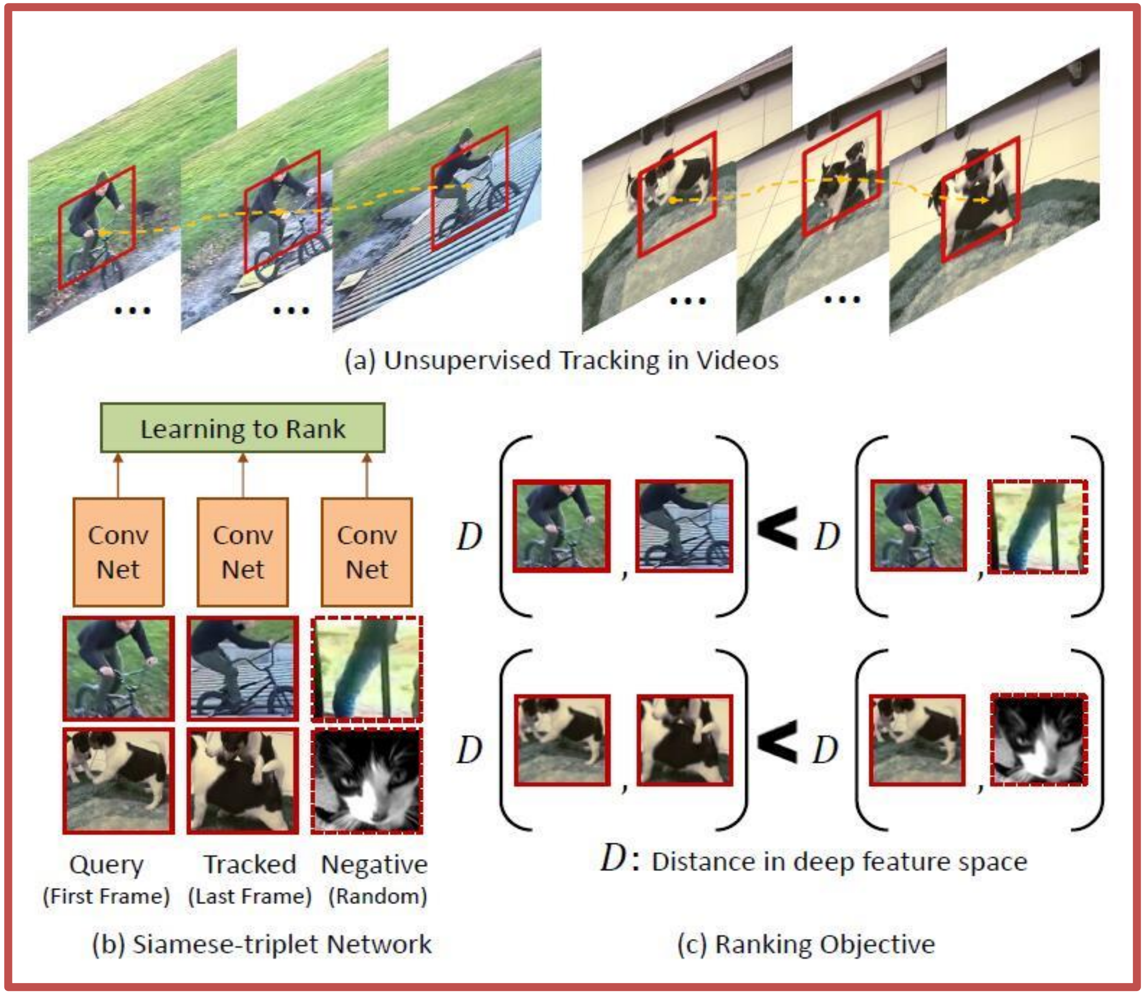
It can match or exceed the performance of supervised models.



Paper: https://arxiv.org/pdf/2002.05709.pdf

UNSUPERVISED LEARNING – OBJECT TRACKING

Visual tracking allows the model to learn about objects without labels.



Paper: http://www.cs.cmu.edu/~xiaolonw/papers/unsupervised_video.pdf

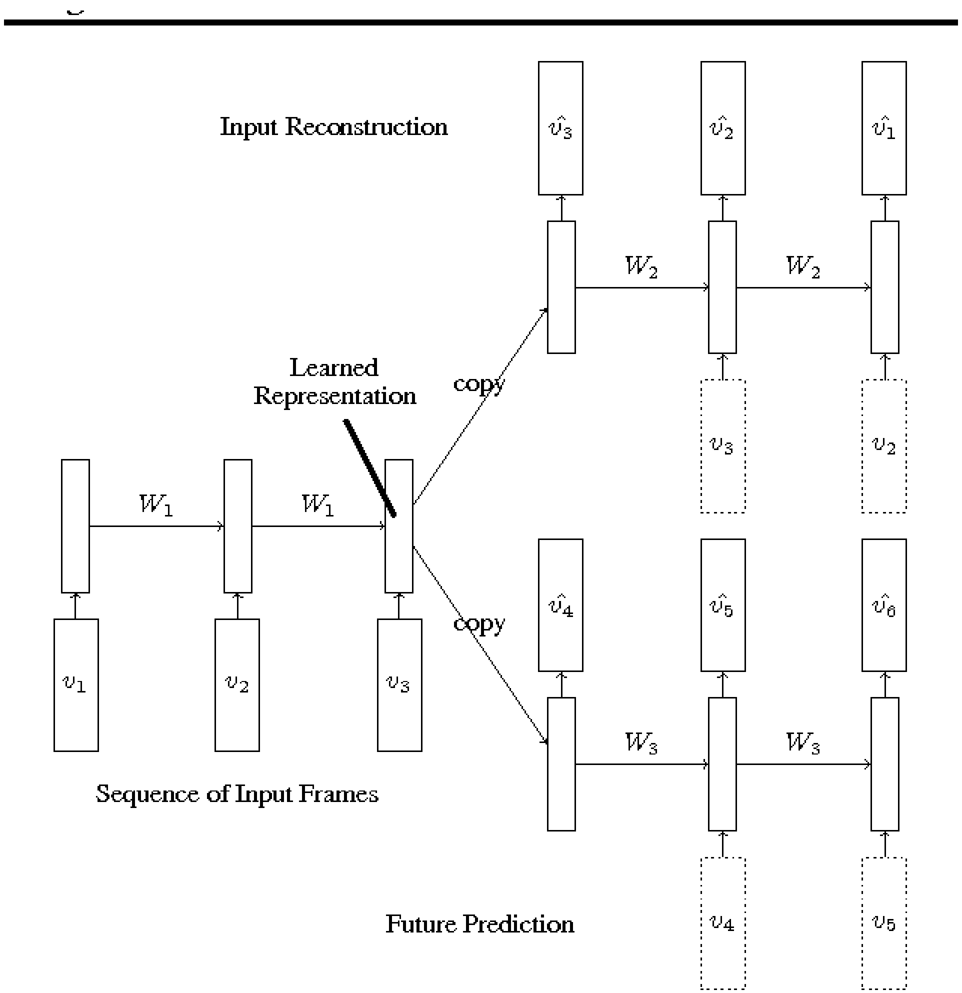
UNSUPERVISED LEARNING – GENERATIVE MODELS

- Generative models can also be useful in learning useful features of an unlabelled dataset
- The model to be used depends on the dataset:
  - GANs and Variational Autoencoders (introduced next lecture) can be useful for learning feature representations of image data by creating new images that look similar to a dataset
  - Future steps in time series data can be predicted using an LSTM or similar and useful features learned in the process (autoregressive modelling).
- As with the autoencoder if a generative model is restricted via training or representation size then it will be forced to learn useful features of the data to predict it

SEMI-SUPERVISED LEARNING

- Semi-supervised learning is used in the case where we have large amounts of unlabelled data and limited labelled data all from the same domain
- An unsupervised learning method is used on the unlabelled data and then the model is fine tuned on the labelled data, like in the transfer learning scenario

SEMI-SUPERVISED LEARNING – EXAMPLE

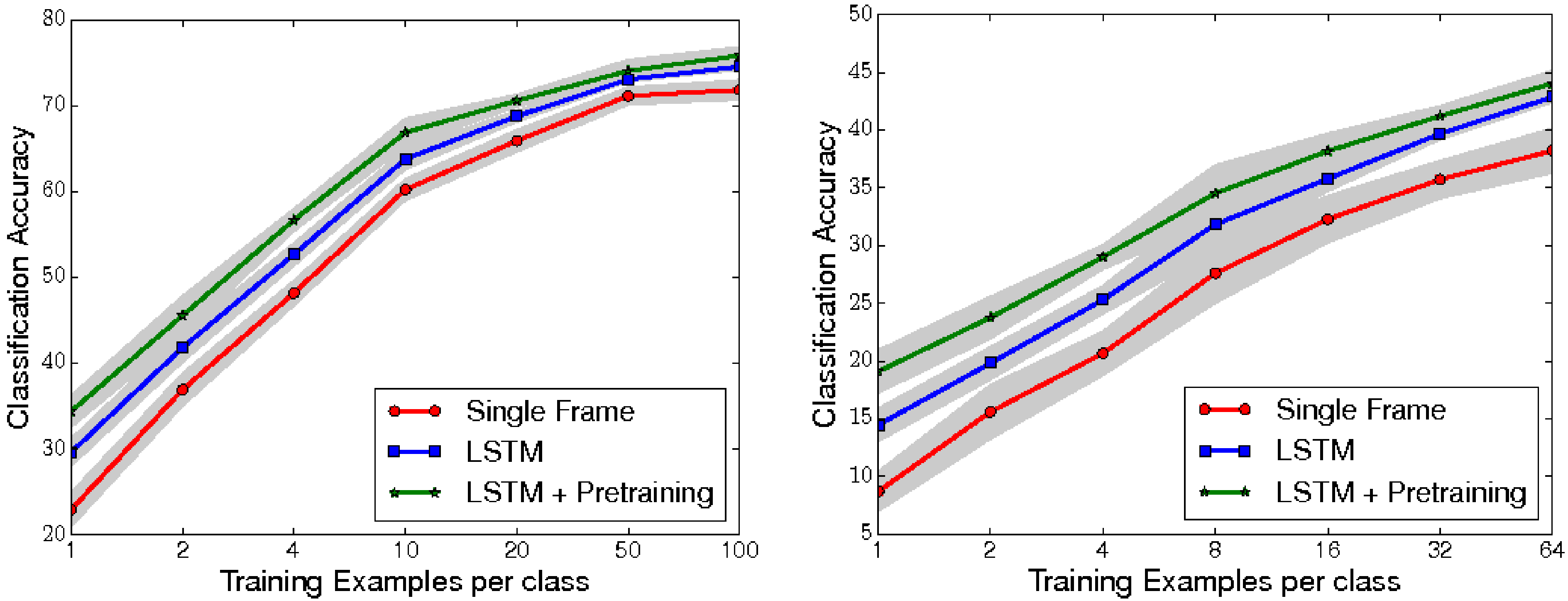


Unsupervised Learning of Video Representations using LSTMs

- Three LSTMs are used. An encoder learns the input representation, then one decoder learns to reconstruct the input (an autoencoder) and the second learns to predict future frames
- Paper: https://pdfs.semanticscholar.org/915f/dd2fdc7880074bd1c1d596f7e7d19ab34e8f.pdf

SEMI-SUPERVISED LEARNING – EXAMPLE

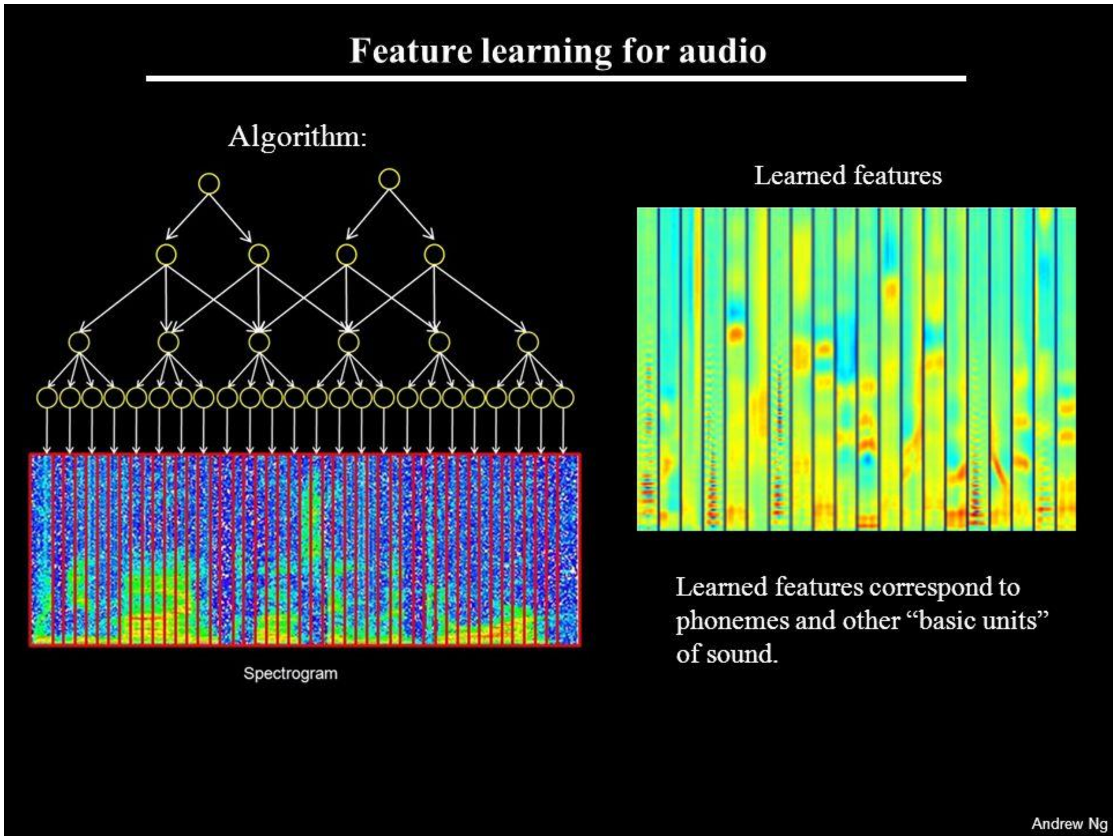
Comparison of an LSTM initialised with the weights from the encoder LSTM on the previous page (green line) compared to the same LSTM with randomly initialised weights. The results show pretraining is particularly important when the number of examples of actions per class are very small



(a) UCF-101 RGB

(b) HMDB-51 RGB

SEMI-SUPERVISED LEARNING – EXAMPLE



Paper: https://arxiv.org/pdf/1206.6471

Unsupervised feature learning trained on audio spectrograms.

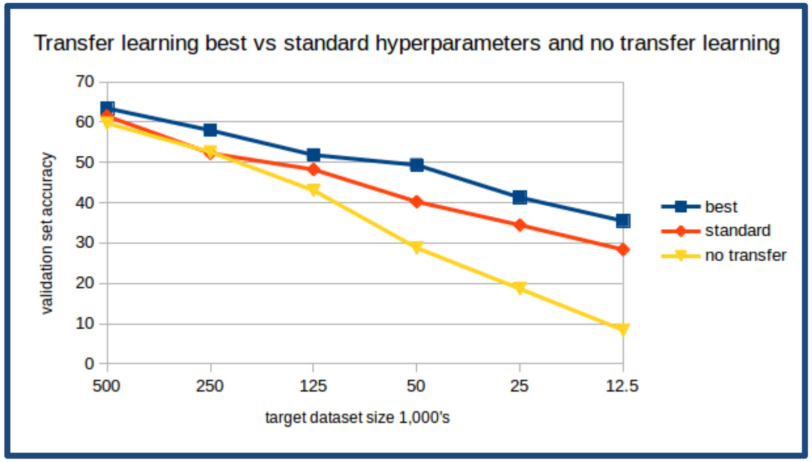
Results show that the learned features correspond to phonemes and other basic units of sound

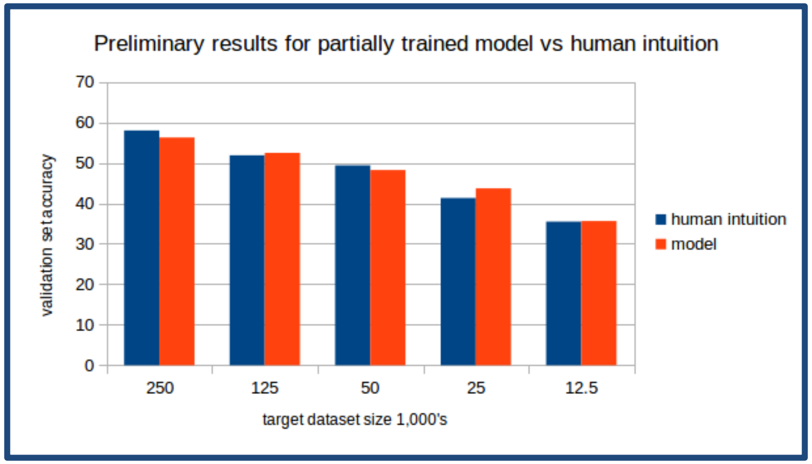
Also that unsupervised pretraining greatly improves classification accuracy on audio classification tasks with limited labelled data.

Note: This research used an older style of neural network called a deep belief network which I won't cover, but the results are still relevant

MY RESEARCH

I'm currently working on the best techniques for doing transfer learning on very small datasets in the image classification area. I'm also working on using DNN models to automate decisions about how to do transfer learning, so people with limited expertise in transfer learning can apply it well.





https://link.springer.com/chapter/10.1007/978-3-030-36708-4_26