

# COMP250: Artificial Intelligence

## 11: Deep Learning

# Deep learning



# Deep learning

# Deep learning

- ▶ Basically, the use of large ANNs with **many layers**

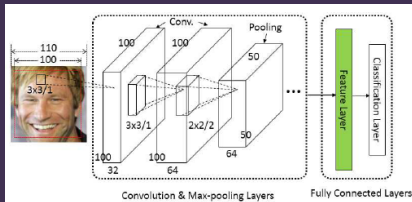
# Deep learning

- ▶ Basically, the use of large ANNs with **many layers**
- ▶ Often uses **large training sets**

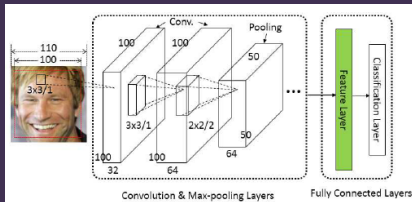
# Deep learning

- ▶ Basically, the use of large ANNs with **many layers**
- ▶ Often uses **large training sets**
- ▶ Training often uses powerful **GPUs** — many times faster than training on the CPU

# Convolutional Neural Networks (ConvNets)



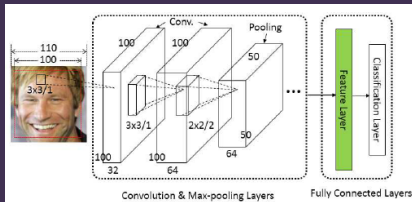
# Convolutional Neural Networks (ConvNets)



- Layers are **2D arrays**

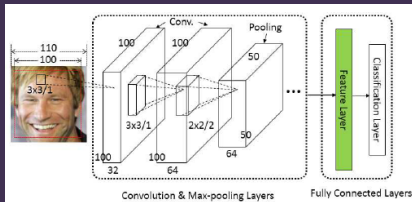


# Convolutional Neural Networks (ConvNets)



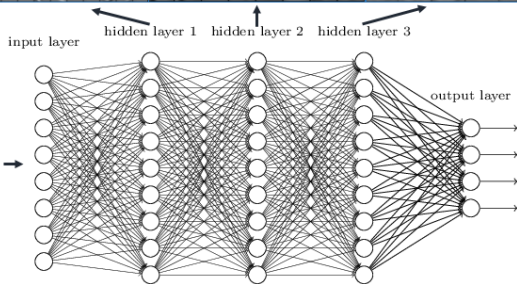
- ▶ Layers are **2D arrays**
- ▶ Neurons in convolutional layers are only connected to nearby neurons

# Convolutional Neural Networks (ConvNets)



- ▶ Layers are **2D arrays**
- ▶ Neurons in convolutional layers are only connected to nearby neurons
- ▶ There are also fully connected layers

Deep neural networks learn hierarchical feature representations



# DeepDream

# DeepDream

- ▶ Train a ConvNet to recognise something (e.g. faces, objects, animals)

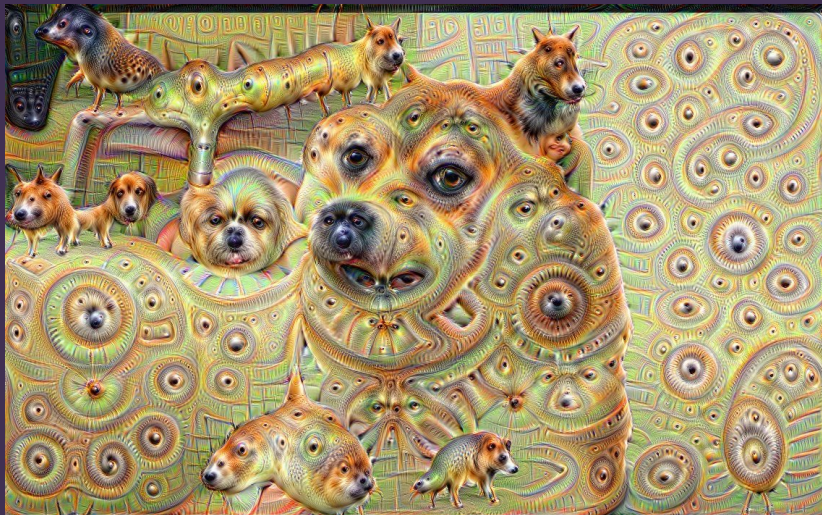
# DeepDream

- ▶ Train a ConvNet to recognise something (e.g. faces, objects, animals)
- ▶ Run the network in “reverse”

# DeepDream

- ▶ Train a ConvNet to recognise something (e.g. faces, objects, animals)
- ▶ Run the network in “reverse”
  - ▶ Adjust the image (e.g. via gradient ascent) so that it is more strongly recognised by the network

# DeepDream





# Style transfer

# Style transfer

- ▶ Train a ConvNet to recognise a particular artistic style

# Style transfer

- ▶ Train a ConvNet to recognise a particular artistic style
- ▶ Run the network in “reverse” on an input image

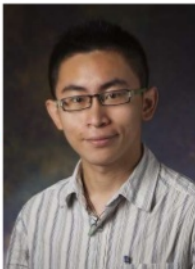
# Style transfer

- ▶ Train a ConvNet to recognise a particular artistic style
- ▶ Run the network in “reverse” on an input image
  - ▶ Adjust the image (e.g. via gradient ascent) so that it is more strongly recognised by the network

## Style transfer



Source image (**Style**)



Target image (**Content**)



Output ([deepart](#))

A Neural Algorithm of Artistic Style [[Gatys et al. 2015](#)]

# Generative Adversarial Networks (GANs)

# Generative Adversarial Networks (GANs)

- ▶ Two ANNs trained in parallel

# Generative Adversarial Networks (GANs)

- ▶ Two ANNs trained in parallel
  - ▶ One to generate “fake” artefacts



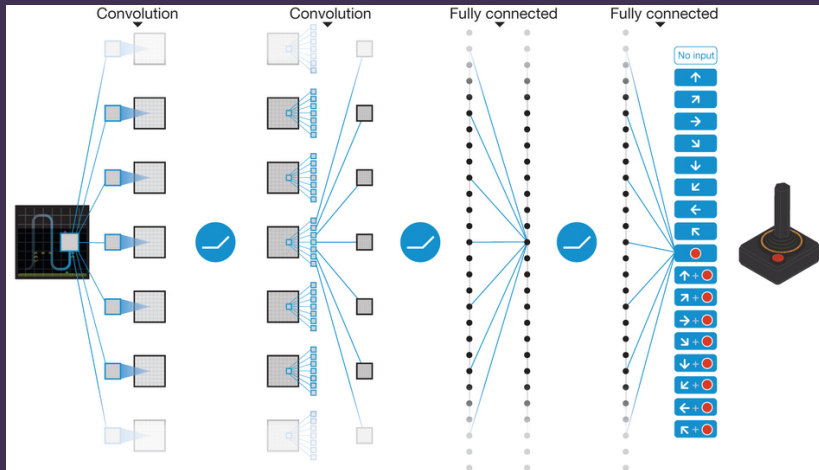
# Generative Adversarial Networks (GANs)

- ▶ Two ANNs trained in parallel
  - ▶ One to generate “fake” artefacts
  - ▶ One to distinguish “real” from “fake”

# Generative Adversarial Networks (GANs)

- ▶ Two ANNs trained in parallel
  - ▶ One to generate “fake” artefacts
  - ▶ One to distinguish “real” from “fake”
- ▶ [http://research.nvidia.com/publication/2017-10\\_Progressive-Growing-of](http://research.nvidia.com/publication/2017-10_Progressive-Growing-of)

# Learning to play Atari games (Mnih et al, 2015)



# AlphaGo (Silver et al, 2017)

# AlphaGo (Silver et al, 2017)

- ▶ MCTS with ANNs for move pruning, simulation playouts and state evaluation

# AlphaGo (Silver et al, 2017)

- ▶ MCTS with ANNs for move pruning, simulation playouts and state evaluation
- ▶ ANNs trained on both expert human matches and self-play (reinforcement learning)

# AlphaGo (Silver et al, 2017)

- ▶ MCTS with ANNs for move pruning, simulation playouts and state evaluation
- ▶ ANNs trained on both expert human matches and self-play (reinforcement learning)
- ▶ Defeated Lee Sedol, world Go champion

# AlphaZero (Silver et al, 2018)



# AlphaZero (Silver et al, 2018)

- ▶ Similar MCTS+ANN architecture to AlphaGo

# AlphaZero (Silver et al, 2018)

- ▶ Similar MCTS+ANN architecture to AlphaGo
- ▶ Trained by reinforcement learning (self-play) only

# AlphaZero (Silver et al, 2018)

- ▶ Similar MCTS+ANN architecture to AlphaGo
- ▶ Trained by reinforcement learning (self-play) only
- ▶ After only 9 hours\* of training, defeated Stockfish (one of the strongest chess programs available) in a 100-match tournament

# AlphaZero (Silver et al, 2018)

- ▶ Similar MCTS+ANN architecture to AlphaGo
- ▶ Trained by reinforcement learning (self-play) only
- ▶ After only 9 hours\* of training, defeated Stockfish (one of the strongest chess programs available) in a 100-match tournament
  - ▶ \* On a cluster of 5000 of Google's custom Tensor Processing Units

# AlphaZero (Silver et al, 2018)

- ▶ Similar MCTS+ANN architecture to AlphaGo
- ▶ Trained by reinforcement learning (self-play) only
- ▶ After only 9 hours\* of training, defeated Stockfish (one of the strongest chess programs available) in a 100-match tournament
  - ▶ \* On a cluster of 5000 of Google's custom Tensor Processing Units
- ▶ Stockfish is based on decades of research by expert chess players and AI programmers — AlphaZero started from no chess-specific knowledge whatsoever (other than the rules of the game)

# Deep learning for PCG

<https://www.youtube.com/watch?v=3wcpLwvBTYo>