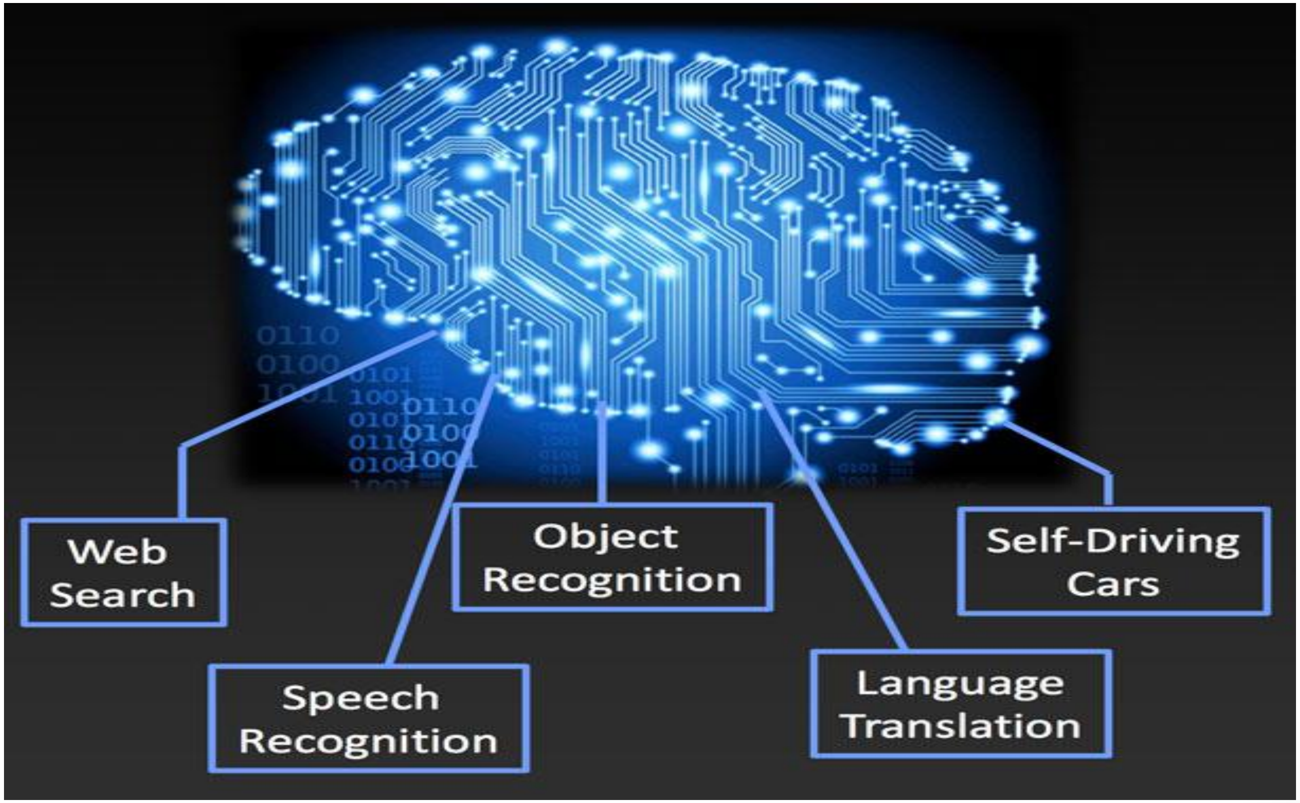
History of Neural Networks and Applications of Historical Convolutional Architectures

By Eugene Huang



Agenda

Part I: History of Neural Networks

Cybernetics (1940-1960s)

Connectionism (1980-1990s)

Deep Learning (2006 – Present)

Part II: Application of Convolutional Architectures

LeNet (1998)

AlexNet (2012)

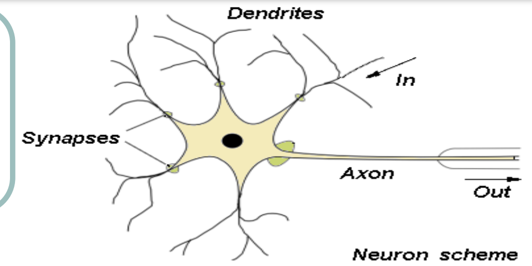
VGG16 (2014)

Part I: History of Neural Networks

Cybernetics: 1940s-1960s

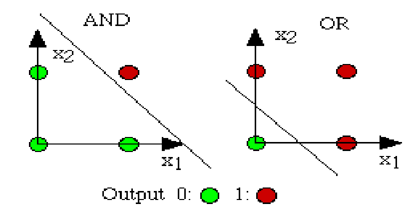
1943

- McCulloch-Pitts neuron: summation over weighted inputs and step activation function



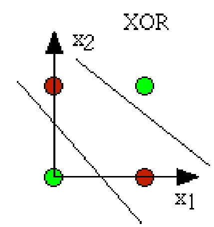
1958

- Mark I Perception by Rosenblatt: included training method to determine optimal weights



1969

- Minsky publishes "Perceptrons" book, showing limitations about linear separability—notably inability to create XOR



1st AI Winter:
Thanks a lot,
Minsky!

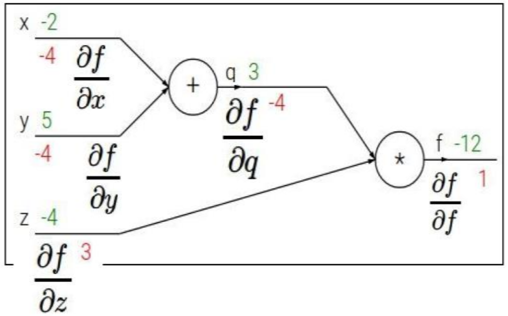
Connectionism: 1980s-1990s

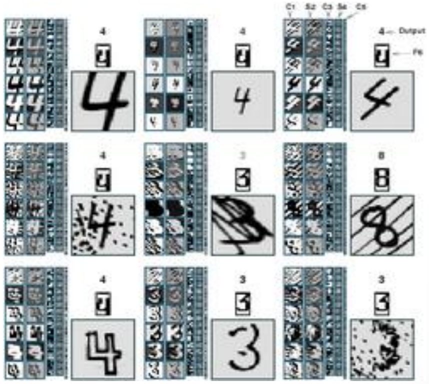
1986
- Hinton et al publishes “Learning Internal Representations by Error Propagation”, essentially popularizes backprop, neural nets are trainable

1989
- “Multilayer Feedforward Networks are Universal Approximators”: with 1 hidden layer, you can do anything!`

1989
- LeCunn publishes “Backpropagation Applied to Handwritten Zip Code Recognition”: invents convolutional layer and pooling

1989
- Precursor to RNN: Waibel et al creates Time-Delay neural networks that take in ordered data





Fun Fact: Geoff Hinton = Great-great-grandson of George Boole
Poor attempt at Chuck Norris joke: Spock, want logic? Go ask Hinton—he was born from it!

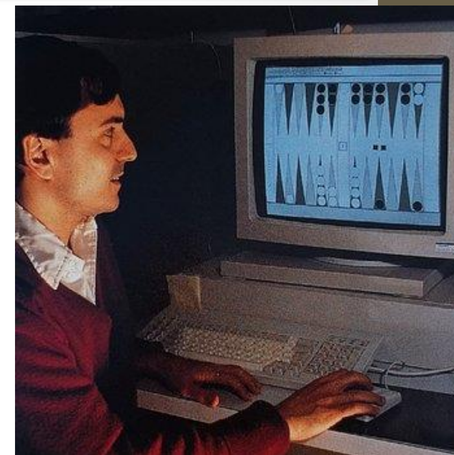
Connectionism: 1980s-1990s (cont'd)

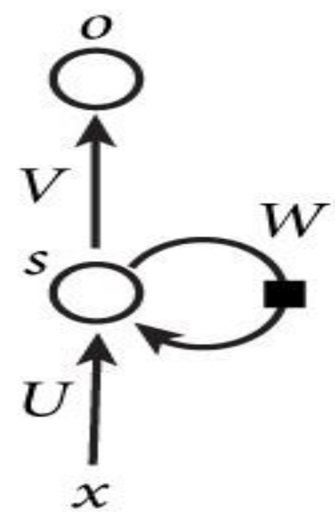
1992
- Reinforcement Learning: TD-Gammon: world class Backgammon AI

1995
- Sebastian Thrun: "Learning to Play the Game of Chess" published, though not competitive with other AI

1997
- Schmidhuber creates LSTM to viably train RNN for long memory, alleviates exploding/vanishing gradient problem

1998
- Yann LeCun develops LeNet--convnet for reading checks





2nd AI Winter: SVMs

Deep Learning: 2006 -Present

- 2006 — Hinton introduced unsupervised pretraining and deep belief nets, showed nets with many layers can be trained *well*
- 2009 — Andrew Ng publishes “Large Scale Deep Unsupervised Learning Using Graphics Processors”, use GPUs
- 2010 — “Understanding the Difficulty of Training Deep Feedforward Neural Networks”, develop ReLu and Xavier initialization
- 2012 — “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups”
- 2012 — “Improving Neural Networks By Prevent Co-Adaption of Feature Detectors”, develop Dropout
- 2012 — “ImageNet Classification with Deep Convolutional Neural Networks”, AlexNet created and significantly outperforms SVMs

3nd AI Winter:
NEVER!

Part II: Application of Convolutional Architectures

Kaggle's Galaxy Zoo

Classify the morphologies of distant galaxies in our Universe

https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge

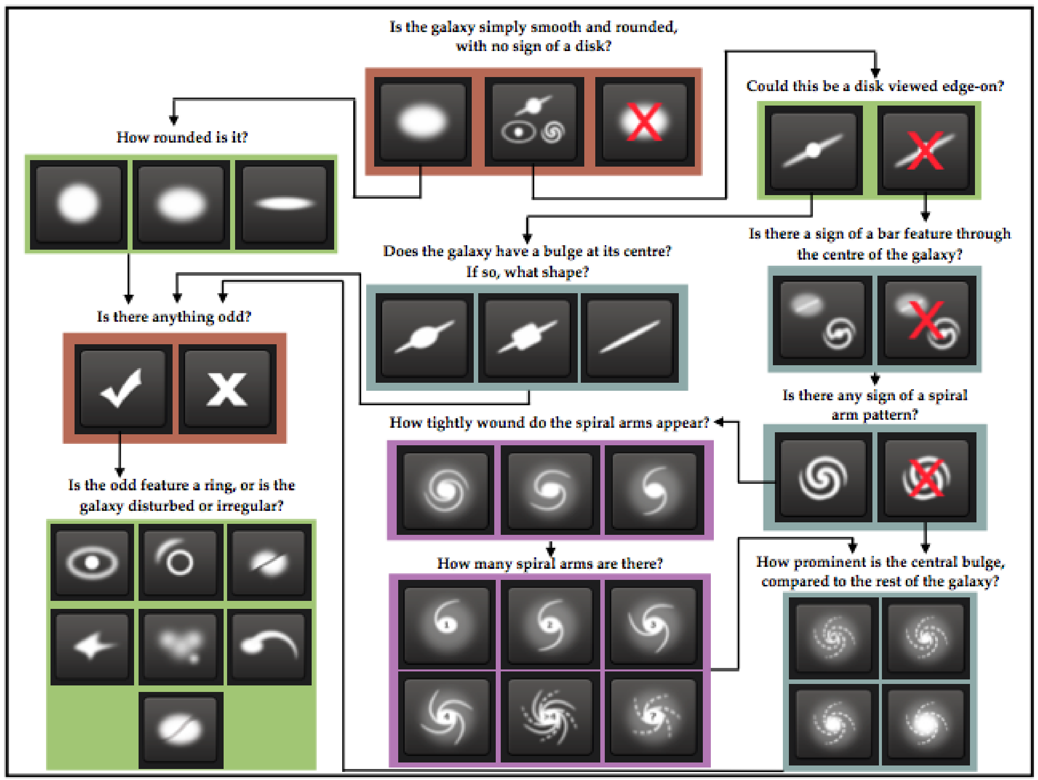


Figure 1. Flowchart of the classification tasks for GZ2, beginning at the top centre. Tasks are colour-coded by their relative depths in the decision tree. Tasks outlined in brown are asked of every galaxy. Tasks outlined in green, blue, and purple are (respectively) one, two or three steps below branching points in the decision tree. Table 2 describes the responses that correspond to the icons in this diagram.

Is the galaxy:

- *Smooth (~43%)*
- *Features or disk (~57%)*
- *Star or artifact? (<0.1%)*

Really a binary classification problem due to imbalanced classes

LeNet (1998)

- Input (1 channel x 32 x 32)
- Convolution (6 filters x 5 x 5)
- Pool (2 x 2)
- Convolution (16 filters x 5 x 5)
- Pool (2 x 2)
- Dense (120 neurons)
- Dense 84 neurons)
- Dense (Softmax into 10)

- Sigmoid Activation
- Not sure if maxpool

LeNet Upgraded

- Input (3 channel x 224 x 224)
- Convolution (32 filters x 5 x 5 with stride 3)
- MaxPool (2 x 2)
- BatchNorm
- Convolution (256 filters x 5 x 5 with stride 3)
- MaxPool (2 x 2)
- BatchNorm
- Dense (640 neurons)
- BatchNorm
- Dense (64 neurons)
- BatchNorm
- Dense (Softmax into 3)

- ReLu Activation
- More filters/neurons for feature extraction from larger image
- “Downsampling” in convolution
- Added BatchNorm

“Gradient-Based Learning Applied to Document Recognition” by Yann LeCun, Yoshua Bengio, et al

AlexNet (2012)

- Input (3 channel x 227 x 227)
- Convolution (96 filters x 11 x 11, stride 4)
- MaxPool (3 x 3 with stride 2)
- Convolution (256 filters x 5 x 5)
- MaxPool (3 x 3 with stride 2)
- Convolution (384 filters x 3 x 3)
- Convolution (384 filters x 3 x 3)
- Convolution (256 filters x 3 x 3)
- MaxPool (3 x 3 with stride 2)
- Dense (4096 neurons)
- Dropout
- Dense (4096 neurons)
- Dropout
- Dense (Softmax into 1000)

- ReLu Activation
- Still high “downsampling” in convolution, large filters
- Used Dropout, not sure with what probability
- Trained on ImageNet using GPUs

AlexNet Upgraded

- BatchNorm (3 channel x 224 x 224)
- Convolution (96 filters x 11 x 11 with stride 4)
- MaxPool (3 x 3 with stride 2 x 2)
- BatchNorm
- Dropout (p = 0.3)
- Convolution (256 filters x 5 x 5)
- MaxPool (3 x 3 stride 2)
- BatchNorm
- Dropout (p = 0.3)
- Convolution (384 filters x 3 x 3)
- BatchNorm
- Convolution (384 x 3 x 3)
- BatchNorm
- Convolution (256 filters x 3 x 3)
- MaxPool (3 x 3 with stride 2)
- BatchNorm
- Dropout (p = 0.5)
- Dense (800 neurons)
- BatchNorm
- Dropout (0.5)
- Dense (50 neurons)
- BatchNorm
- Dropout (p = 0.5)
- Dense (Softmax into 3)

- ReLu Activation
- Added ***More*** Dropout -> Reduce Overfitting
- Added BatchNorm -> Decrease Training Time

“ImageNet Classification with Deep Convolutional Neural Networks” by Geoff Hinton, Ilya Sutskever, et al

VGG16 (2014)

- Input (3 channel x 224 x 224)

- Convolution (64 filters x 3 x 3)
- Convolution (64 filters x 3 x 3)
- MaxPool (2 x 2)

- Convolution (128 filters x 3 x 3)
- Convolution (128 filters x 3 x 3)
- MaxPool (2 x 2)

- Convolution (256 filters x 3 x 3)
- Convolution (256 filters x 3 x 3)
- Convolution (256 filters x 3 x 3)
- MaxPool (2 x 2)

- Convolution (512 filters x 3 x 3)
- Convolution(512 filters x 3 x 3)
- Convolution (512 filters x 3 x 3)
- MaxPool (2 x 2)

- Convolution (512 filters x 3 x 3)
- Convolution (512 filters x 3 x 3)
- Convolution (512 filters x 3 x 3)
- MaxPool (2 x 2)

- Dense (4096 neurons)
- Dropout (p = 0.5)
- Dense (4096 neurons)
- Dropout (p = 0.5)

- Dense (Softmax into 1000)

- Built from Convolutional Block and Fully Connected Block

"Very Deep Convolutional Networks for Large-Scale Image Recognition"
by Karen Simonyan and Andrew Zisserman. ***Non-Canadians***, gasp!

VGG16 Upgraded

- Input (3 channel x 224 x 224)
- Convolution (64 filters x 3 x 3)
- Convolution (64 filters x 3 x 3)
- MaxPool (2 x 2)
- Convolution (128 filters x 3 x 3)
- Convolution (128 filters x 3 x 3)
- MaxPool (2 x 2)

- Convolution (256 filters x 3 x 3)
- Convolution (256 filters x 3 x 3)
- Convolution (256 filters x 3 x 3)
- MaxPool (2 x 2)
- Convolution (512 filters x 3 x 3)
- Convolution(512 filters x 3 x 3)
- Convolution (512 filters x 3 x 3)
- MaxPool (2 x 2)

- Convolution (512 filters x 3 x 3)
- Convolution (512 filters x 3 x 3)
- Convolution (512 filters x 3 x 3)
- MaxPool (2 x 2)
- Dense (4096 neurons)
- **BatchNorm**
- Dropout (p = 0.5)
- Dense (4096 neurons)
- **BatchNorm**
- Dropout (p = 0.5)
- Dense (Softmax into 3)

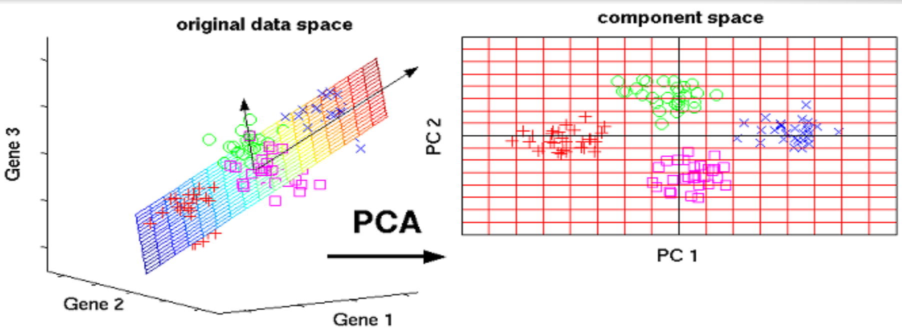
- Added BatchNorm
- Added Data Augmentation
- Used Pre-trained weights

- Split convolution and dense layers into separate layers for faster training time

One More Thing… Random Forest

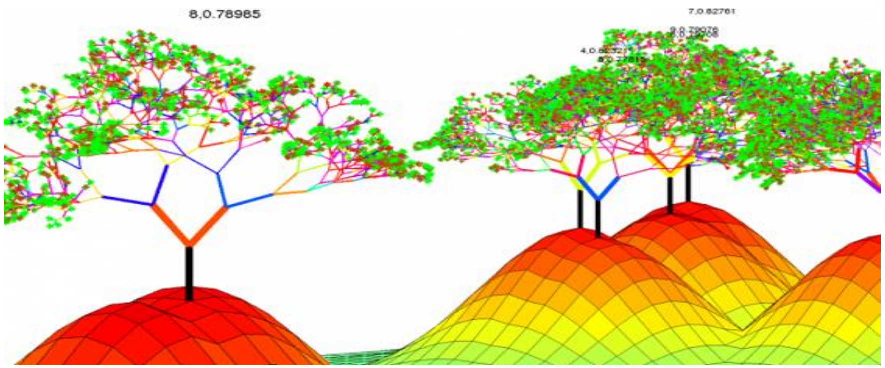
PCA

- Incremental PCA
- Iterative Method for RAM efficiency
- 224 * 224 * 3 = 150,528 dimensions => 100 components



Random Forest

- GridSearched over RF hyperparameters
- Used Stratified K-Fold Cross-Validation



Neural Net

- Input
- Dropout + BatchNorm
- Dense
- Dropout + BatchNorm
- Dense (Softmax)

Lessons Learned | Interesting Thoughts

- Don't be a Hero! Use a pre-trained net
- PCA + RF are actually very good
- Neural nets are powerful but are still a manual process
- Linear Algebra for forward pass, calculus for backpropagation
- Convolutions take time; Dense layers take RAM

- Neural net = Infinitely flexible function + All-purpose parameter fitting (Backprop) + Fast and scalable (GPU)
- Old architectures weren't bad, just hardware wasn't good (enough)
- Nets don't really mimic the brain; Brains don't do backprop
- Though CapsuleNet actually have a biological inspiration
- Never count the Canadians out!

Thank You!