Review

Nan Ye

School of Mathematics and Physics
The University of Queensland

A Much Sought-after Technology

0

By **Alison DeNisco Rayome** in **CXO**

on February 5, 2019, 8:39 AM PST

Jobs in AI and machine learning are exploding, as countries race to develop the emerging technology, according to a UiPath report.

By Stephen Zafarino, Contributor, CIO | JULY 27, 2018 06:30 AM PT

Opinions expressed by ICN authors are their own.

OPINION

The outlook for machine learning in tech: ML and AI skills in high demand

22,034 views | Mar 17, 2019, 10:35am

Machine Learning Engineer Is The Best Job In The U.S. According To Indeed

Good News for Job Seekers With Machine Learning Skills: There is a Shortage of Talent

A short pool of AI-trained job seekers has slowed down hiring and impeded growth at some companies

Stacy Stanford in Data Driven Investor Follow

Oct 20, 2018 · 7 min read ★

Course Objectives

Learn basic theories, algorithms and models of machine/deep learning and be able to apply them.

- Understand and explain the intuition, ideas and theory of deep learning algorithms and models.
- Assess whether a deep learning algorithm is effective and appropriate for an application.
- Propose suitable deep learning solutions and implement them for real world problems.
- Effectively explain deep learning solutions in the form of oral presentations and reports.

Have fun...

Machine Learning

- Machine learning is about problem solving with data.
- We only see some examples, but need to find something that works on new examples.
- Typically formulated as optimizing an unknown expected performance measure.
- We often optimize a performance measure estimated using data.

How to Apply Machine Learning Well

https://xkcd.com/1838/

- There is no simple recipe for making machine learning work
- Bad practice
  - Treat learning algorithms as a blackbox for turning data into answers to your question.
- Good practice
  - Understand the domain
    - what information is useful and need to be captured?
    - can you solve the problem yourself?
  - Understand the algorithms
    - machine learning algorithms are often tricky to debug, and you need good understanding of the algorithms when something goes wrong

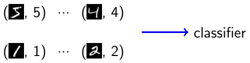
Building blocks

- Week 1-2: machine learning basics
  - *overview, regression, classification, PCA, learning theory, model selection*
- Week 3-4: neural network basics
  - *Perceptron, Adaline, Hopfield, gradient-based learning, MLPs, autograd and PyTorch*
- Week 5-7: deep architectures
  - *CNNs, RNNs*
- Week 7 - 8: optimization
  - *difficulties, initialization, normalization, adaptive learning rates*
- Week 8-11: improving generalization
  - *model selection, model averaging, regularization, residual learning, adversarial learning, attention*
- Week 11 - 12: unsupervised learning
  - *autoencoders, VAEs, GANs*
- Week 12 (17 May - 19 May): reinforcement learning
  - *MDP, planning, Q-learning, SARSA, policy gradient*

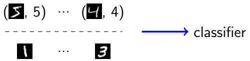
Machine Learning Basics

Some learning problems

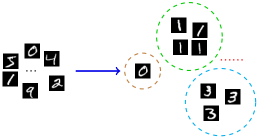
- Supervised learning



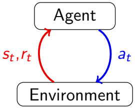
- Semi-supervised learning



- Unsupervised learning



- Reinforcement learning



Regression

- OLS
- Ridge regression
- Basis function method
- Regression function
- Nearest neighbor regression
- Kernel regression

Classification

- Decision boundary
- Nearest neighbor
- Naive Bayes
- Logistic regression
- SVMs

Other topics

- Principal component analysis
- Statistical learning theory
- Model selection

Basics of Neural Networks

Perceptron and Adaline

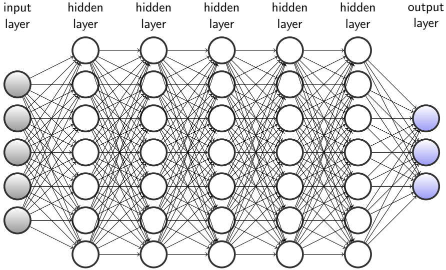
- Simple feedforward networks but works well on simple problems
- A number of important ideas: convergence analysis, stochastic learning, surrogate loss

Hopfield networks

- An interesting class of recurrent neural network
- Conceptually complex as compared to Perceptron and Adaline

MLP

- Deep learning = feature learning + classifier learning



Gradient-based learning

- Gradients are used to update model parameters
- Gradient computation: numerical gradient, symbolic differentiation, autodiff
- PyTorch

Deep Architectures

CNNs, RNNs, ResNets

- Deep networks are hard to learn
  - complex loss surface, vanishing/exploding gradients
- Structural priors are useful
  - CNNs: local receptive field, shared weights, sub-sampling
  - LSTM: memory cell
  - ResNet: learn residuals

Optimization Tricks

- Initialization: diversity, stability
- Input normalization, batch normalization
- Adaptive learning rates

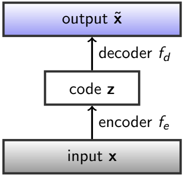
Improving Generalization

- To improve generalization, we make our optimization problem more similar to optimizing the expected performance.
- This can be done by making changes to the data, objective function, model family.
- Some commonly used methods
  - Model selection
  - Model averaging
  - Regularization: data augmentation, ℓ_1/ℓ_2 regularization, early stopping, dropout.
  - Residual learning
  - Adversarial learning
  - Structural priors (e.g. attention)

Unsupervised learning

Autoencoders

- Using autoencoders for learning data representation/dimension reduction/denoising



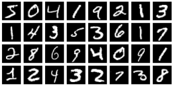
Generative modelling

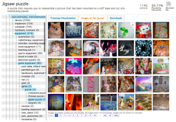
- Variational autoencoders
  - Learns a generative distribution for data via latent variables (autoencoders don't)
  - Can be used to generate examples similar to seen ones
- GAN
  - Generative modelling as a game
  - Generates higher-quality data as compared to VAE

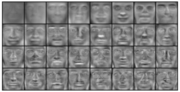
Reinforcement Learning

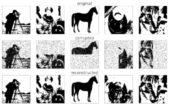
- Modelling interaction with stochastic environments as an MDP
- Planning: value iteration
- Reinforcement learning: Q-learning, SARSA, policy gradient.

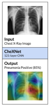
Applications

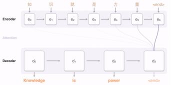


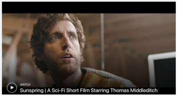


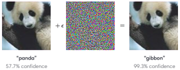


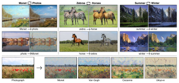


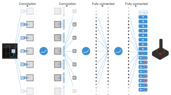












and many others...

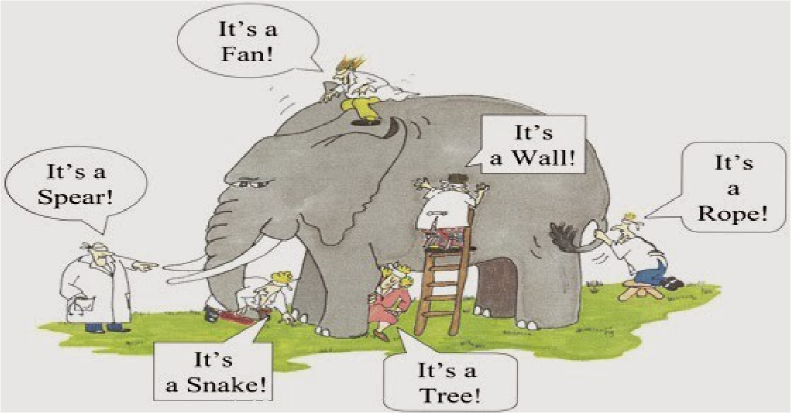
Final Remarks

There isn't a single NN recipe that solves all your problems!

Final Remarks

There isn't a single NN recipe that solves all your problems!

- *Try traditional ML methods — they are good baselines, sometimes working better.*
- *Exploit NNs' modularity and rich design space to mix-and-match and be creative!*
  - *know the math — you may be just one equation away from being creative*
- *A good solution often involve complex trade-offs*
  - *amount of data, model complexity, computing power, ...*



We still have a lot to know about deep learning!

- *Many engineering successes, but theoretical understanding is far from complete.*
- *Perhaps, you will contribute to the practice and theory of DL/ML/AI?!*