Artificial Intelligence

Deep Learning and Reinforcement Learning

1131AI06
MBA, IM, NTPU (M5276) (Fall 2024)
Tue 2, 3, 4 (9:10-12:00) (B3F17)

https://meet.google.com/paj-zhhj-mya

Min-Yuh Day, Ph.D.
Professor

Institute of Information Management, National Taipei University

https://web.ntpu.edu.tw/~myday

2024-11-12

Syllabus

Week	Date	Subject/Topics
1	2024/09/10	Introduction to Artificial Intelligence
2	2024/09/17	Mid-Autumn Festival (Day off)
3	2024/09/24	Artificial Intelligence and Intelligent Agents; Problem Solving
4	2024/10/01	Knowledge, Reasoning and Knowledge Representation; Uncertain Knowledge and Reasoning
5	2024/10/08	Case Study on Artificial Intelligence I
6	2024/10/15	Machine Learning: Supervised and Unsupervised Learning

Syllabus

Week	Date	Subject/Topics
7	2024/10/22	The Theory of Learning and Ensemble Learning
8	2024/10/29	Midterm Project Report
9	2024/11/05	Self-Learning
10	2024/11/12	Deep Learning and Reinforcement Learning
11	2024/11/19	Case Study on Artificial Intelligence II
12	2024/11/26	Deep Learning for Natural Language Processing

Syllabus

Week	Date	Subject/Topics
13	2024/12/03	Computer Vision and Robotics
14	2024/12/10	Generative AI, Philosophy and Ethics of AI and the Future of AI
15	2024/12/17	Final Project Report I
16	2024/12/24	Final Project Report II

Deep Learning and Reinforcement Learning

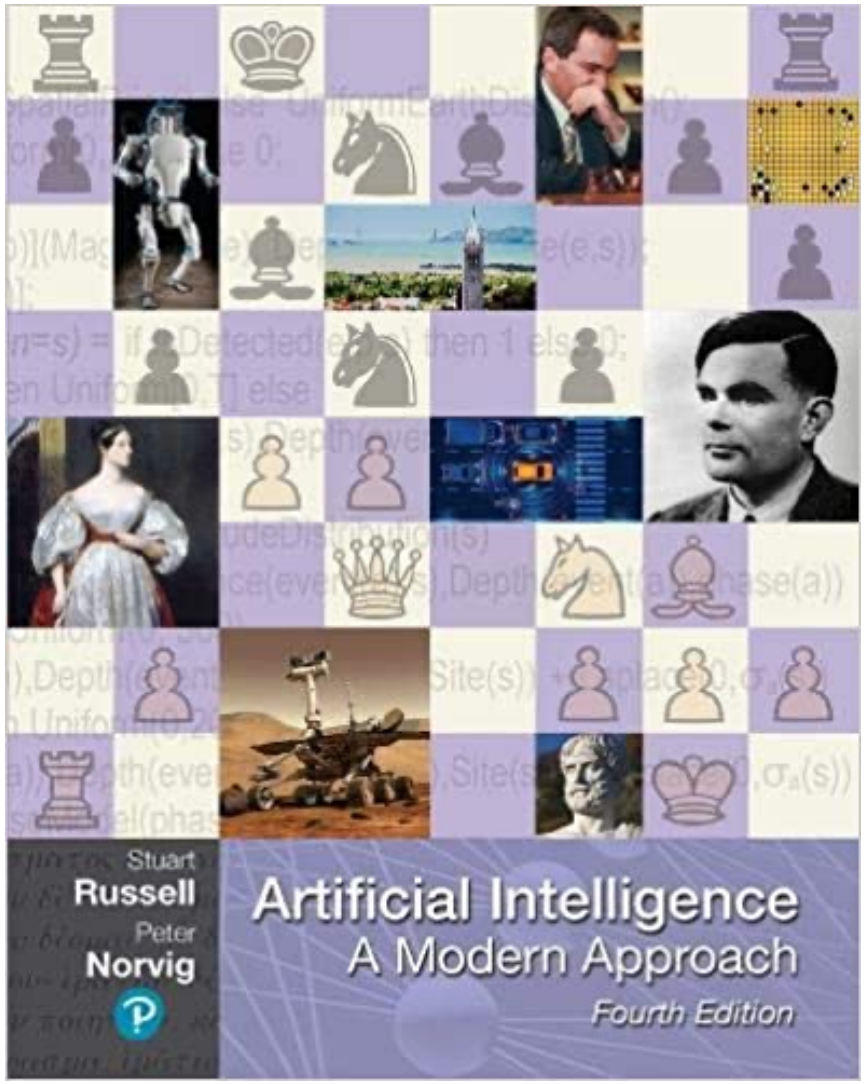
Outline

- Deep Learning (DL)
  - Neural Networks (NN)
  - Convolutional Neural Networks (CNN)
  - Recurrent Neural Networks (RNN)
- Reinforcement Learning (RL)
  - Markov Decision Processes (MDP)
  - Deep Reinforcement Learning (DRL) Algorithms
    - SARSA
    - Q-Learning
    - DQN, A3C, Rainbow

Stuart Russell and Peter Norvig (2020),

Artificial Intelligence: A Modern Approach,

4th Edition, Pearson



Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

https://www.amazon.com/Artificial-Intelligence-A-Modern-Approach/dp/0134610997/

Artificial Intelligence: A Modern Approach

1. Artificial Intelligence
2. Problem Solving
3. Knowledge and Reasoning
4. Uncertain Knowledge and Reasoning
5. Machine Learning
6. Communicating, Perceiving, and Acting
7. Philosophy and Ethics of AI

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Artificial Intelligence:
Machine Learning

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Artificial Intelligence:
5. Machine Learning

- Learning from Examples
- Learning Probabilistic Models
- Deep Learning
- Reinforcement Learning

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Artificial Intelligence:
Reinforcement Learning

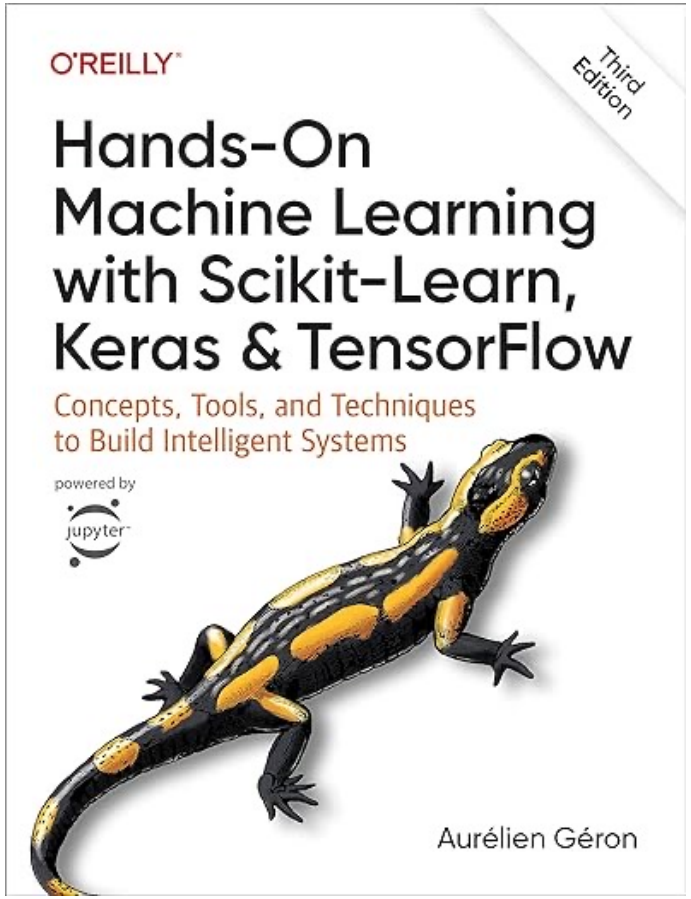
- Learning from Rewards
- Passive Reinforcement Learning
- Active Reinforcement Learning
- Generalization in Reinforcement Learning
- Policy Search
- Apprenticeship and Inverse Reinforcement Learning
- Applications of Reinforcement Learning

Aurélien Géron (2022),

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow:

Concepts, Tools, and Techniques to Build Intelligent Systems,

3rd Edition, O'Reilly Media



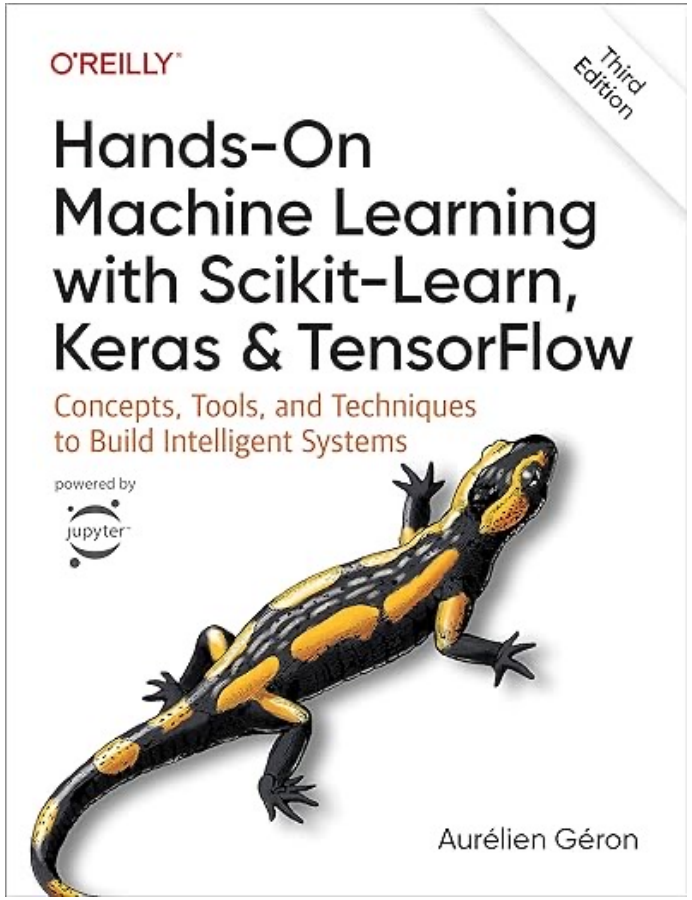
https://github.com/ageron/handson-ml3

Source: https://www.amazon.com/Hands-Machine-Learning-Scikit-Learn-TensorFlow/dp/1098125975

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

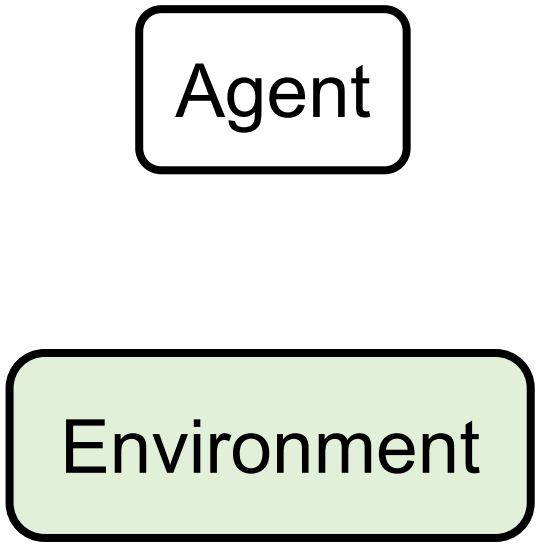
Notebooks

1. The Machine Learning landscape
2. End-to-end Machine Learning project
3. Classification
4. Training Models
5. Support Vector Machines
6. Decision Trees
7. Ensemble Learning and Random Forests
8. Dimensionality Reduction
9. Unsupervised Learning Techniques
10. Artificial Neural Nets with Keras
11. Training Deep Neural Networks
12. Custom Models and Training with TensorFlow
13. Loading and Preprocessing Data
14. **Deep Computer Vision Using Convolutional Neural Networks**
15. **Processing Sequences Using RNNs and CNNs**
16. Natural Language Processing with RNNs and Attention
17. Autoencoders, GANs, and Diffusion Models
18. **Reinforcement Learning**
19. Training and Deploying TensorFlow Models at Scale



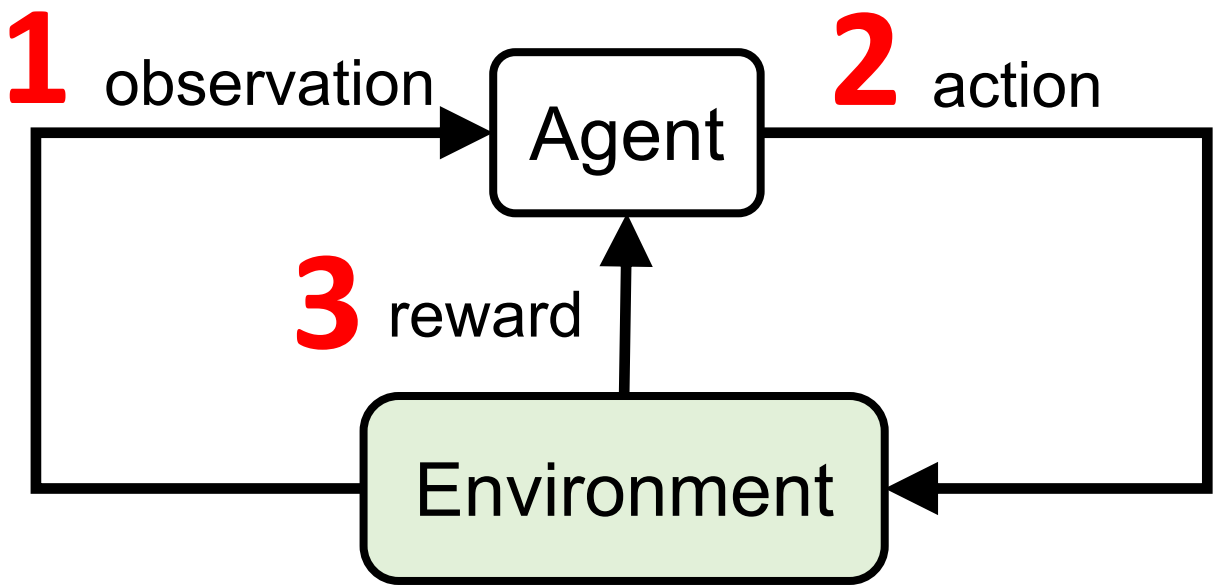
https://github.com/ageron/handson-ml3

Reinforcement Learning (DL)



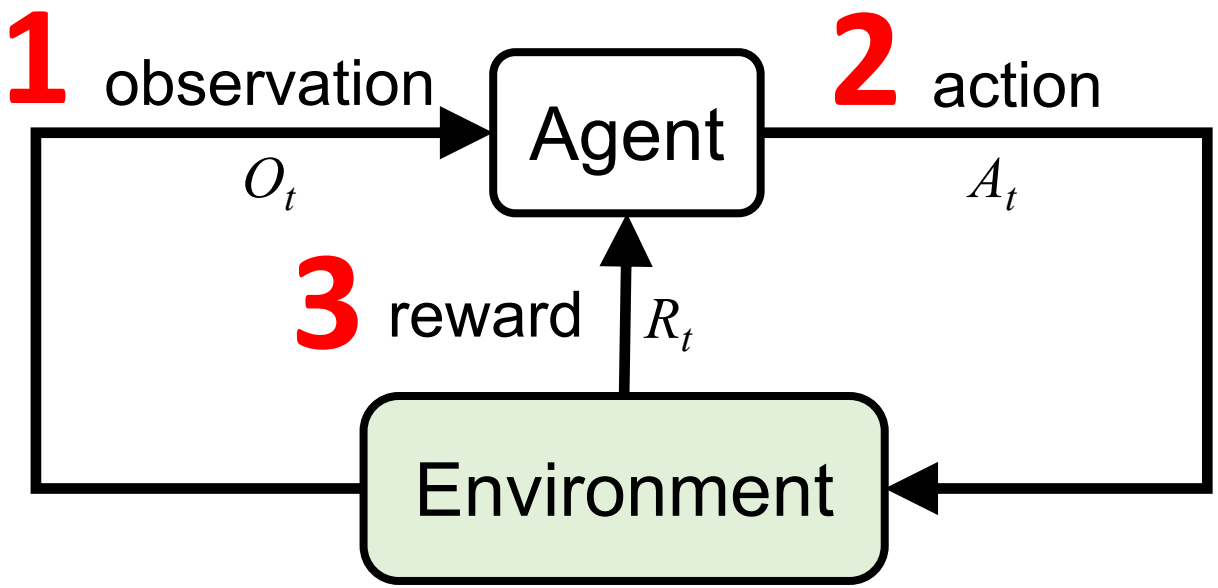
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)



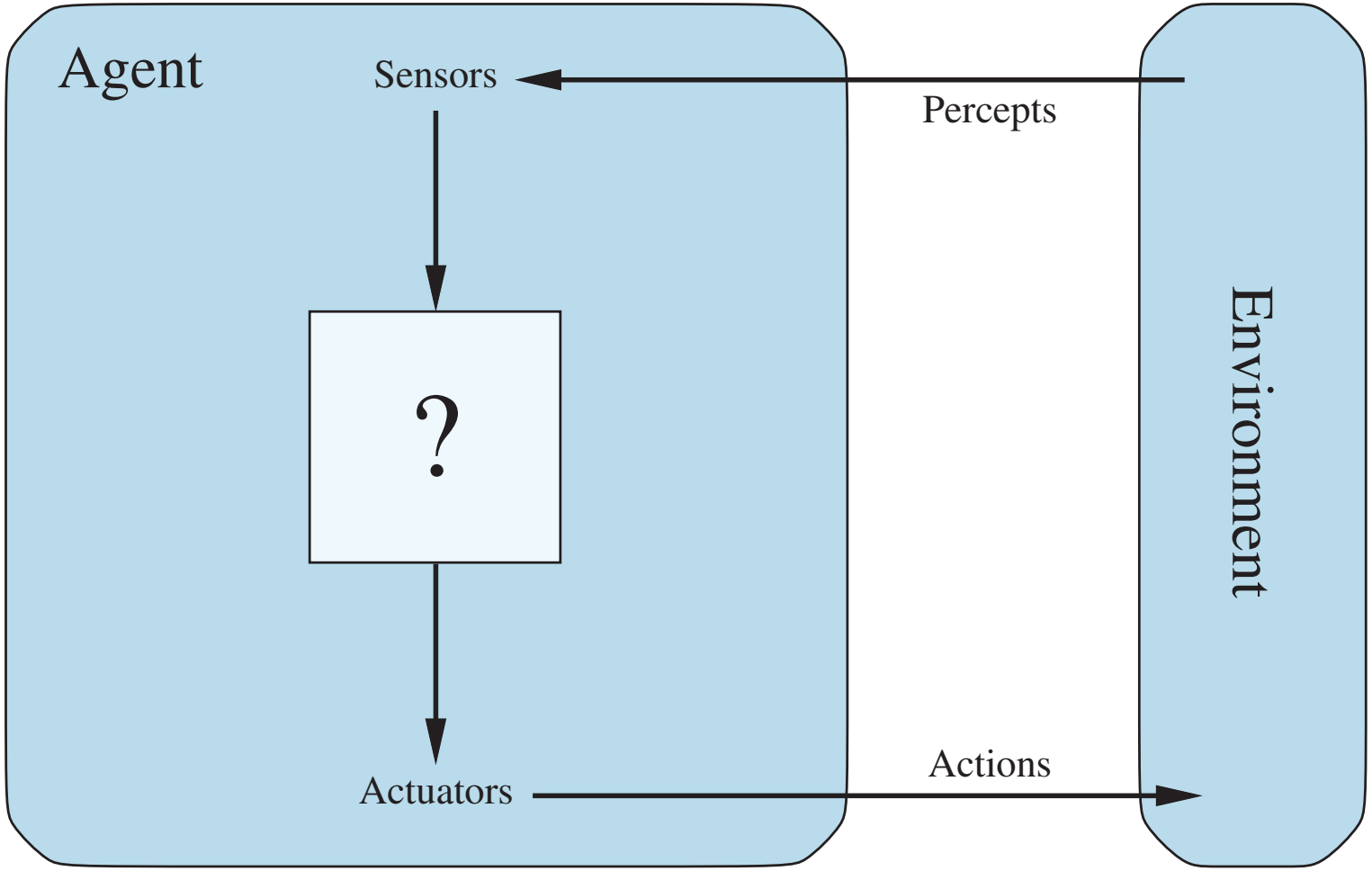
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)



Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Agents interact with environments through sensors and actuators



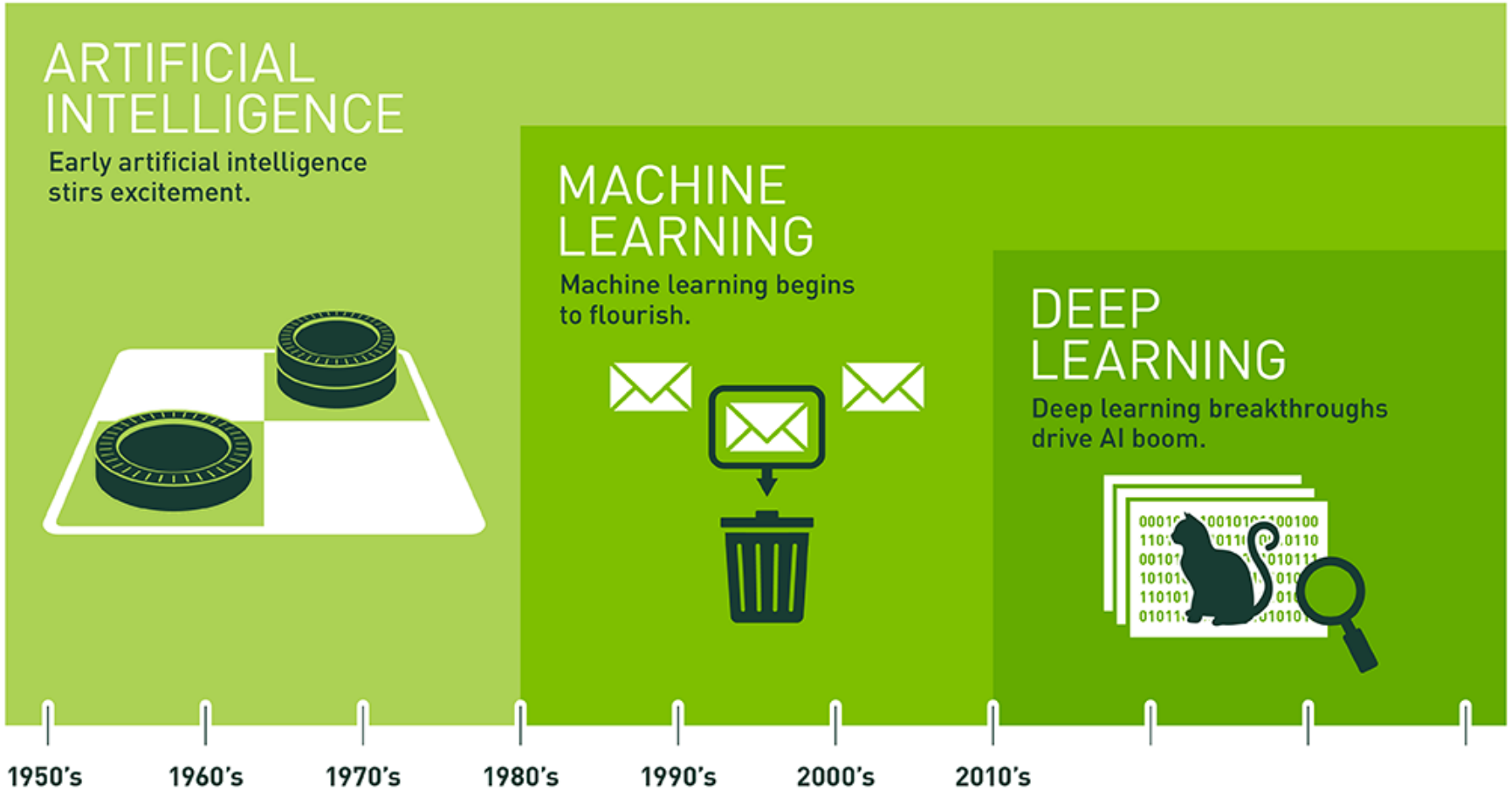
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

AI Acting Humanly:
The Turing Test Approach
(Alan Turing, 1950)

- Knowledge Representation
- Automated Reasoning
- Machine Learning (ML)
  - Deep Learning (DL)
- Computer Vision (Image, Video)
- Natural Language Processing (NLP)
- Robotics

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

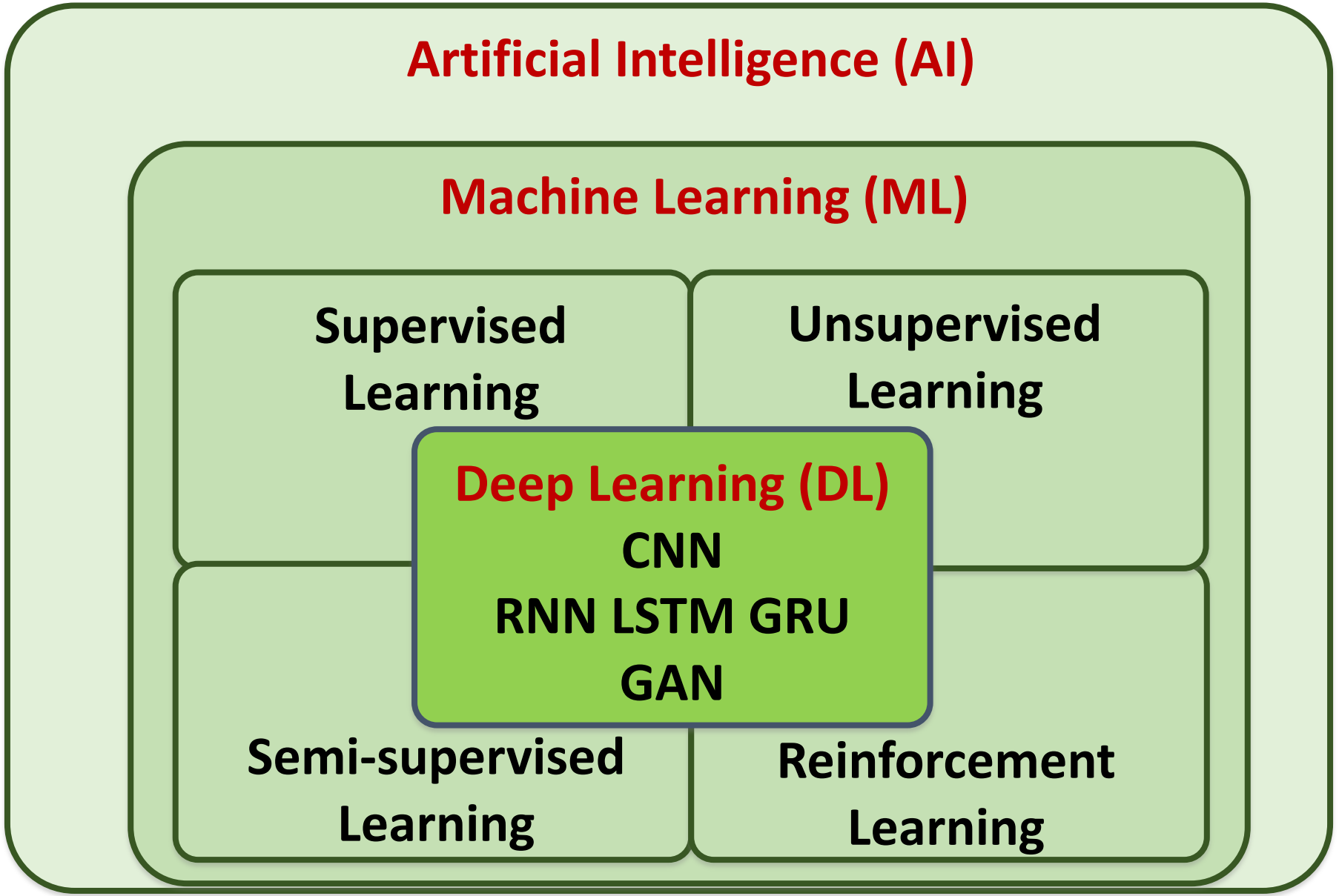
Artificial Intelligence
Machine Learning & Deep Learning



Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

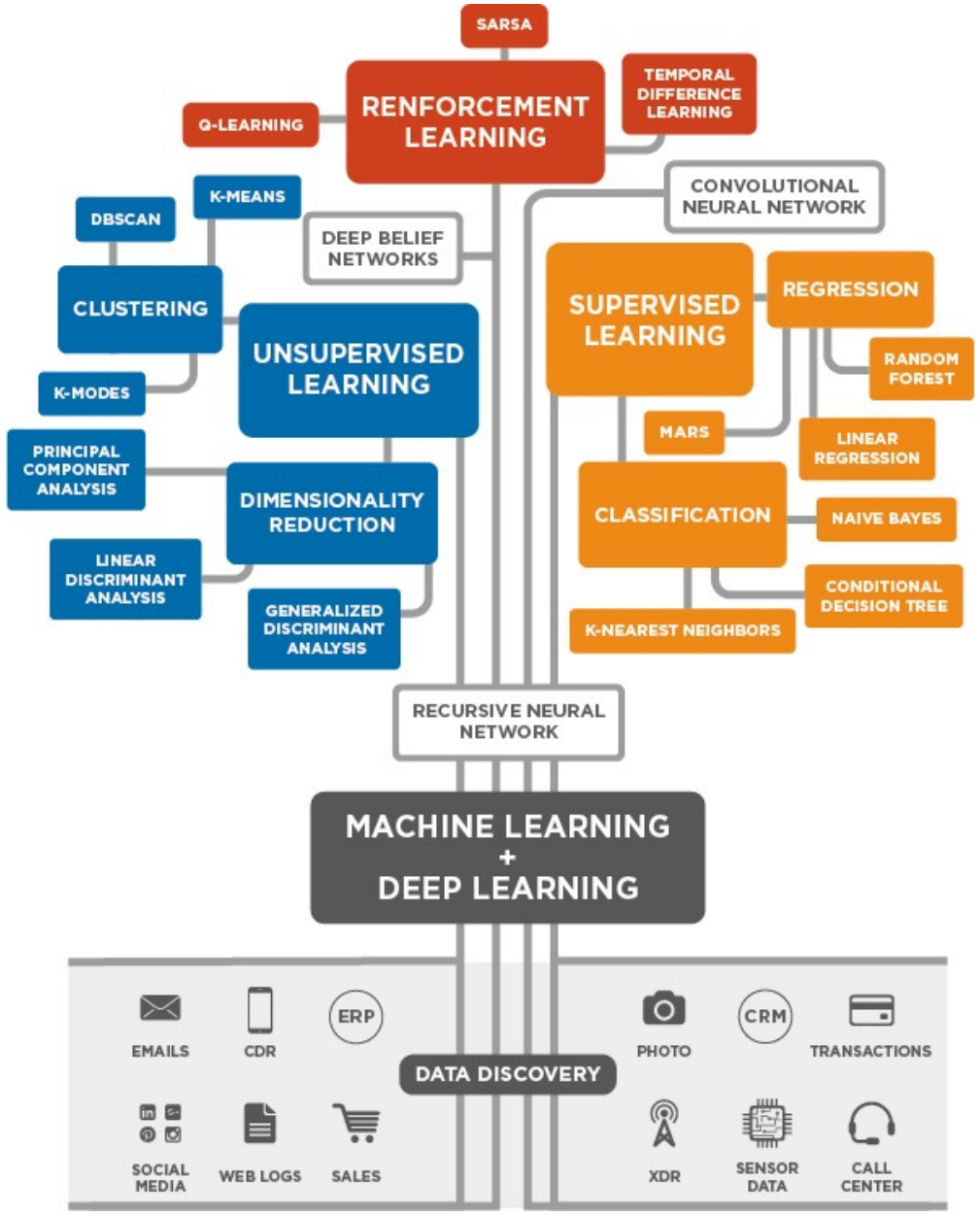
Source: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/

AI, ML, DL



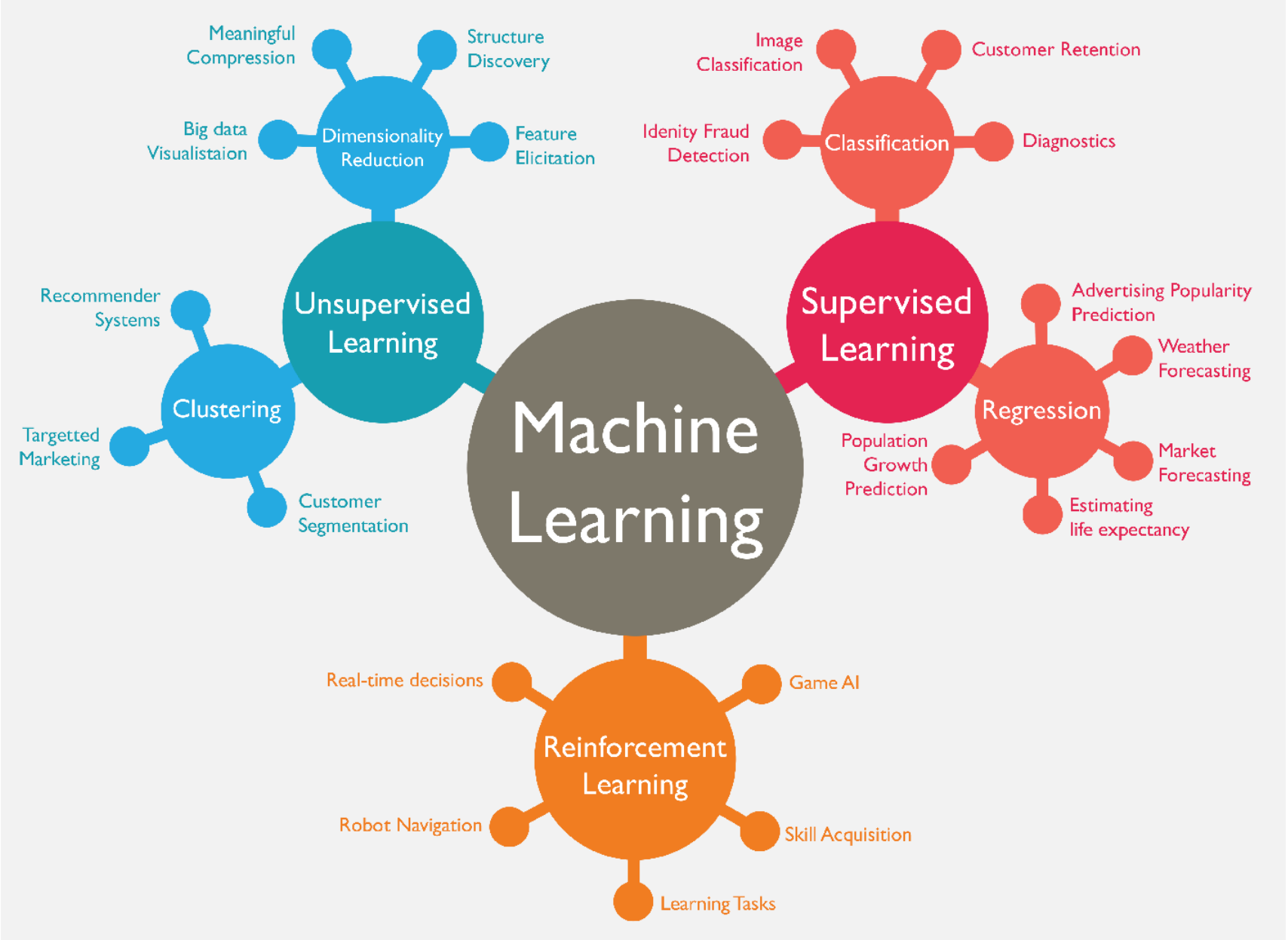
Source: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/deep_learning.html

3 Machine Learning Algorithms



Source: Enrico Galimberti, http://blogs.teradata.com/data-points/tree-machine-learning-algorithms/

Machine Learning (ML)



Source: https://www.mactores.com/services/aws-big-data-machine-learning-cognitive-services/

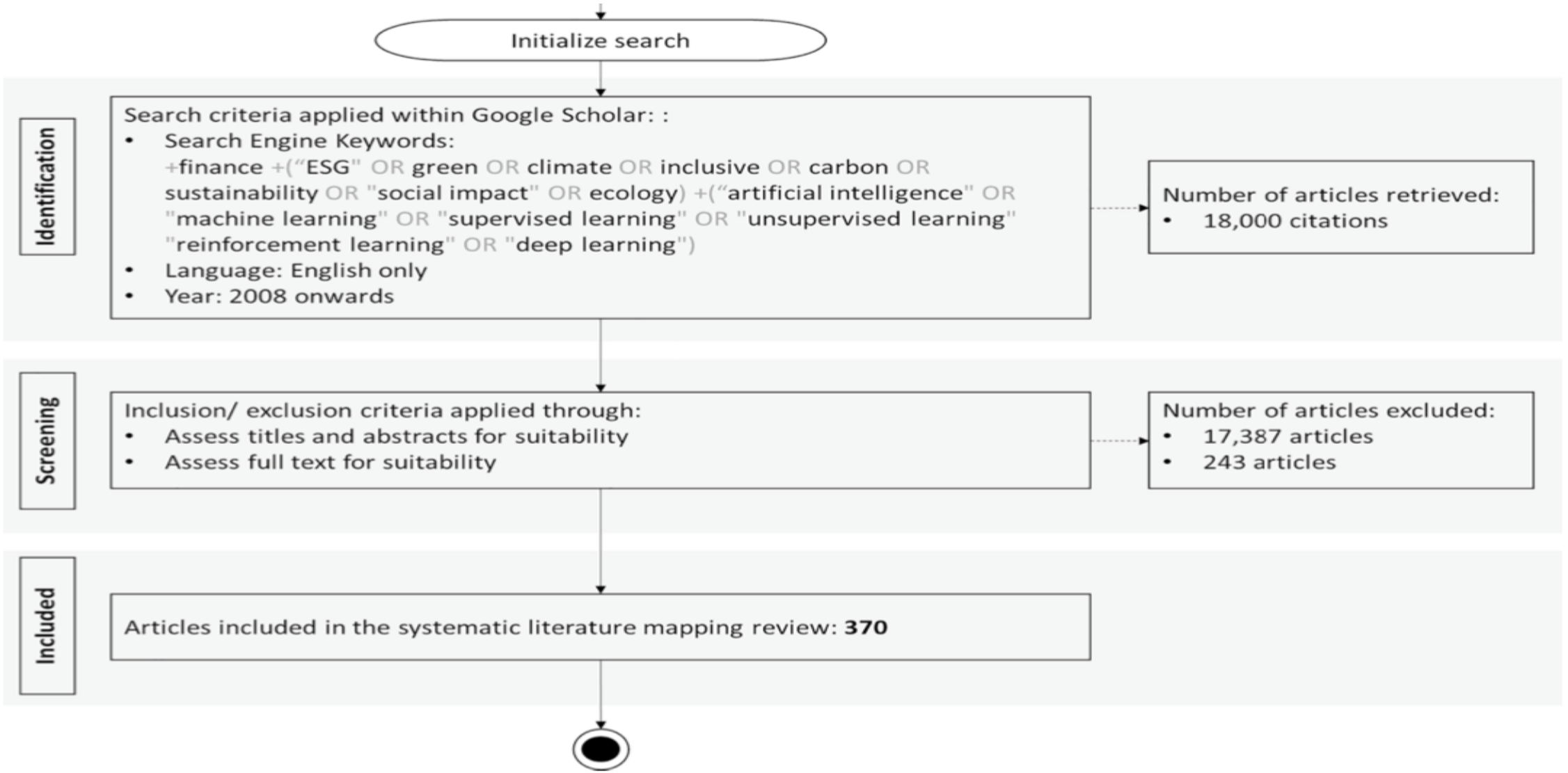
Tristan Lim (2024).

"Environmental, social, and governance (ESG) and artificial intelligence in finance: State-of-the-art and research takeaways."

Artificial Intelligence Review

57, no. 4 (2024): 1-64.

ESG and Artificial Intelligence in Finance
PRISMA—the systematic mapping process



Source: Tristan Lim (2024). "Environmental, social, and governance (ESG) and artificial intelligence in finance: State-of-the-art and research takeaways." Artificial Intelligence Review 57, no. 4 (2024): 1-64.

Network Analysis of ESG and AI in Finance

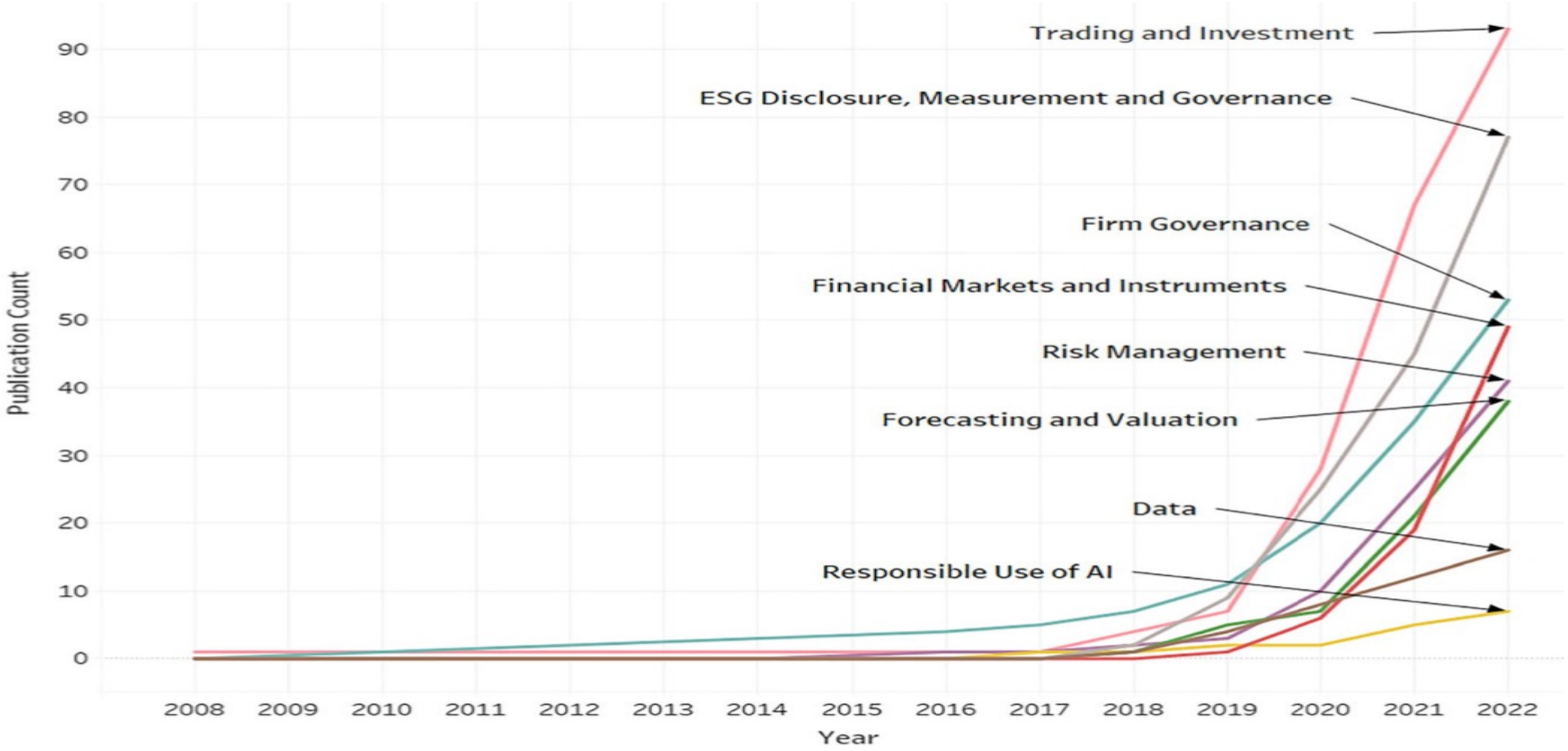
Source: Tristan Lim (2024). "Environmental, social, and governance (ESG) and artificial intelligence in finance: State-of-the-art and research takeaways." Artificial Intelligence Review 57, no. 4 (2024): 1-64.

Research Area of ESG and AI in Finance

Thematic research area	Total paper count	Total citation count	Citation impact
Trading and investment	91	1578	17.3
ESG disclosure, measurement and governance	77	809	10.5
Firm governance	52	1082	20.8
Financial markets and instruments	50	542	10.8
Risk management	41	1030	25.1
Forecasting and valuation	38	178	4.7
Data	14	206	14.7
Responsible use of AI	7	324	46.3
Overall	370	5749	15.5

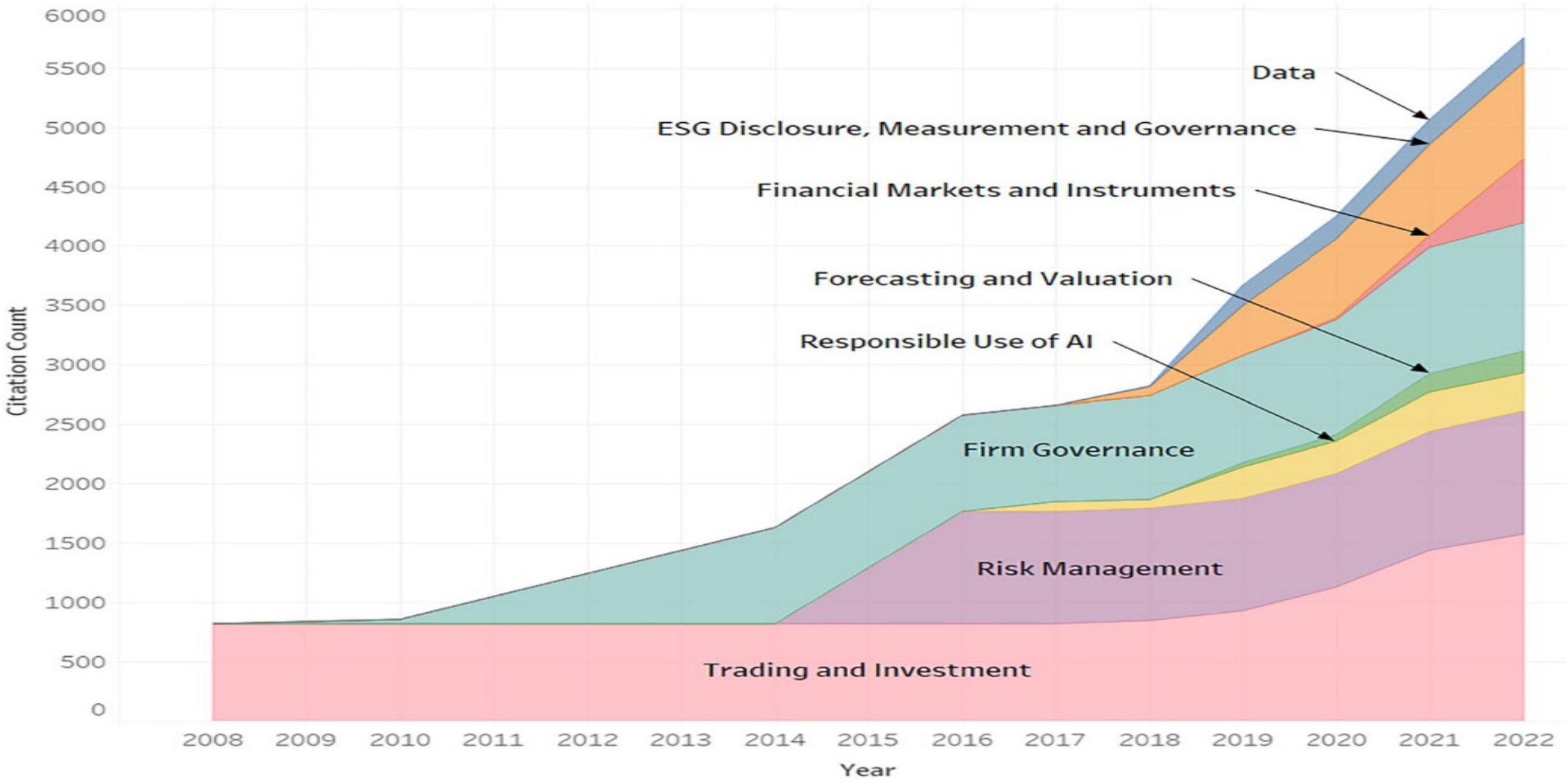
Source: Tristan Lim (2024). "Environmental, social, and governance (ESG) and artificial intelligence in finance: State-of-the-art and research takeaways." Artificial Intelligence Review 57, no. 4 (2024): 1-64.

Total publication count by research thematic areas between 2008 and 2022



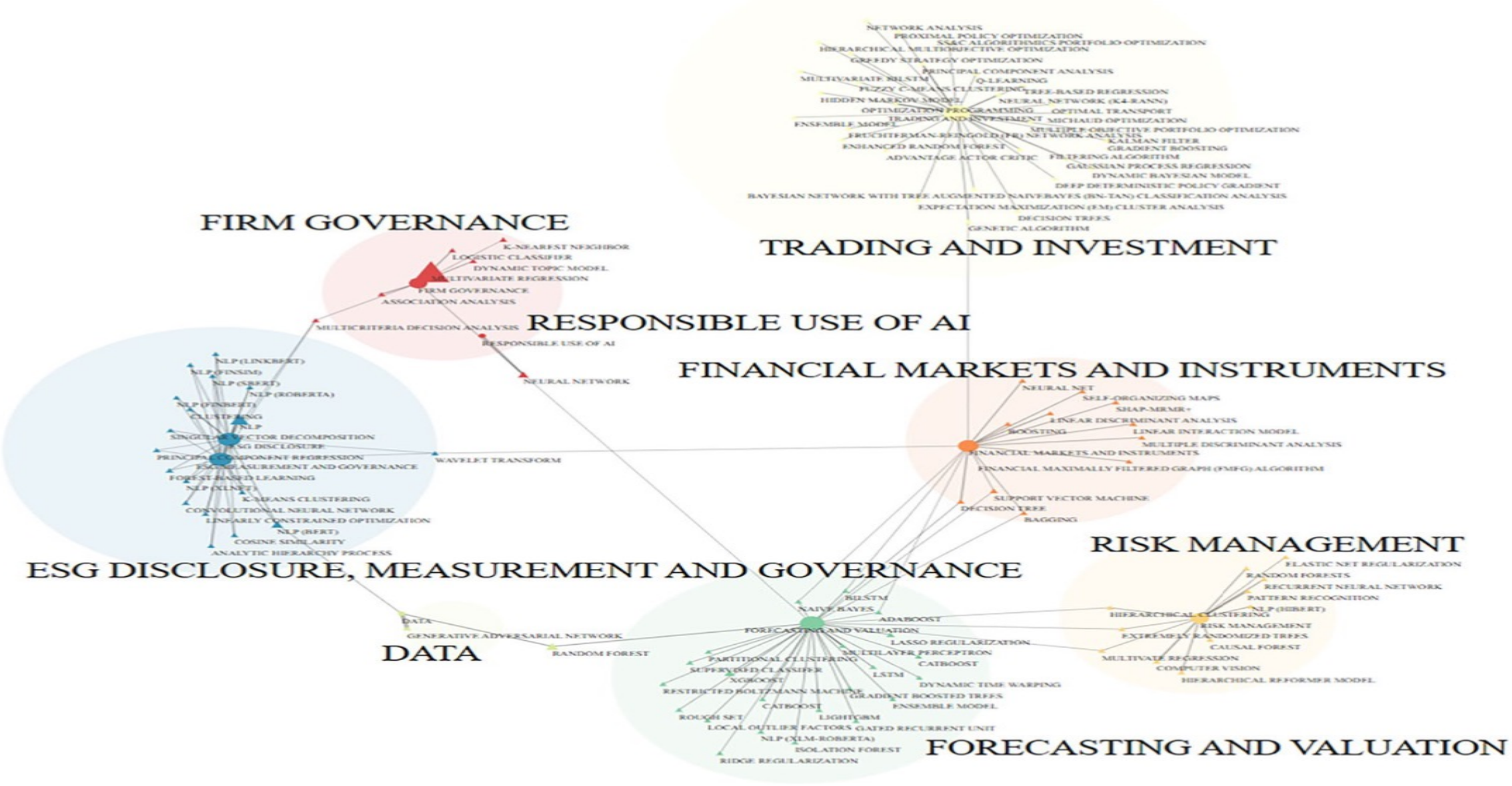
Source: Tristan Lim (2024). "Environmental, social, and governance (ESG) and artificial intelligence in finance: State-of-the-art and research takeaways." Artificial Intelligence Review 57, no. 4 (2024): 1-64.

Total citation count by research thematic areas between 2008 and 2022



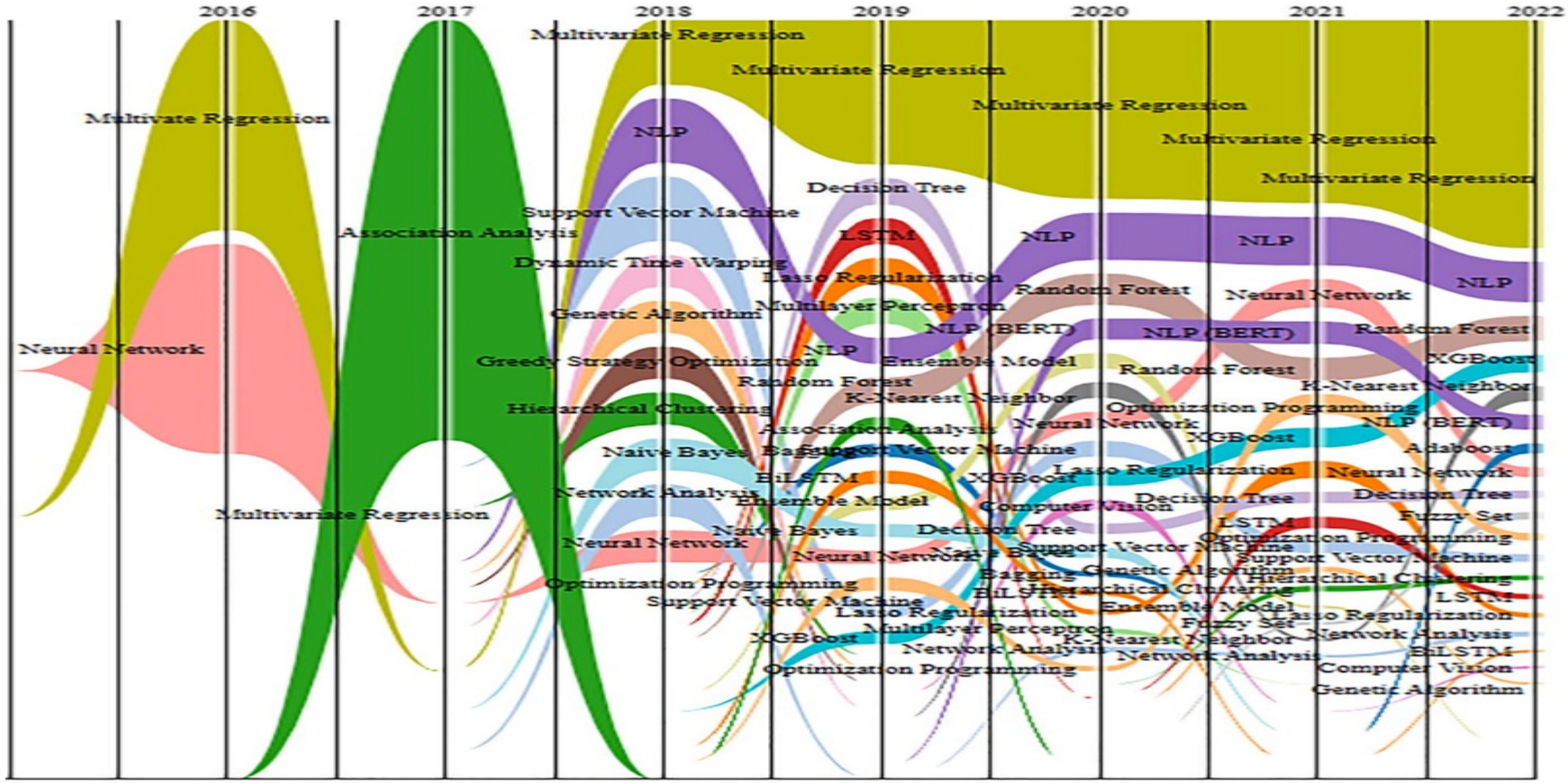
Source: Tristan Lim (2024). "Environmental, social, and governance (ESG) and artificial intelligence in finance: State-of-the-art and research takeaways." Artificial Intelligence Review 57, no. 4 (2024): 1-64.

Network Analysis on Research Domains and AI Techniques



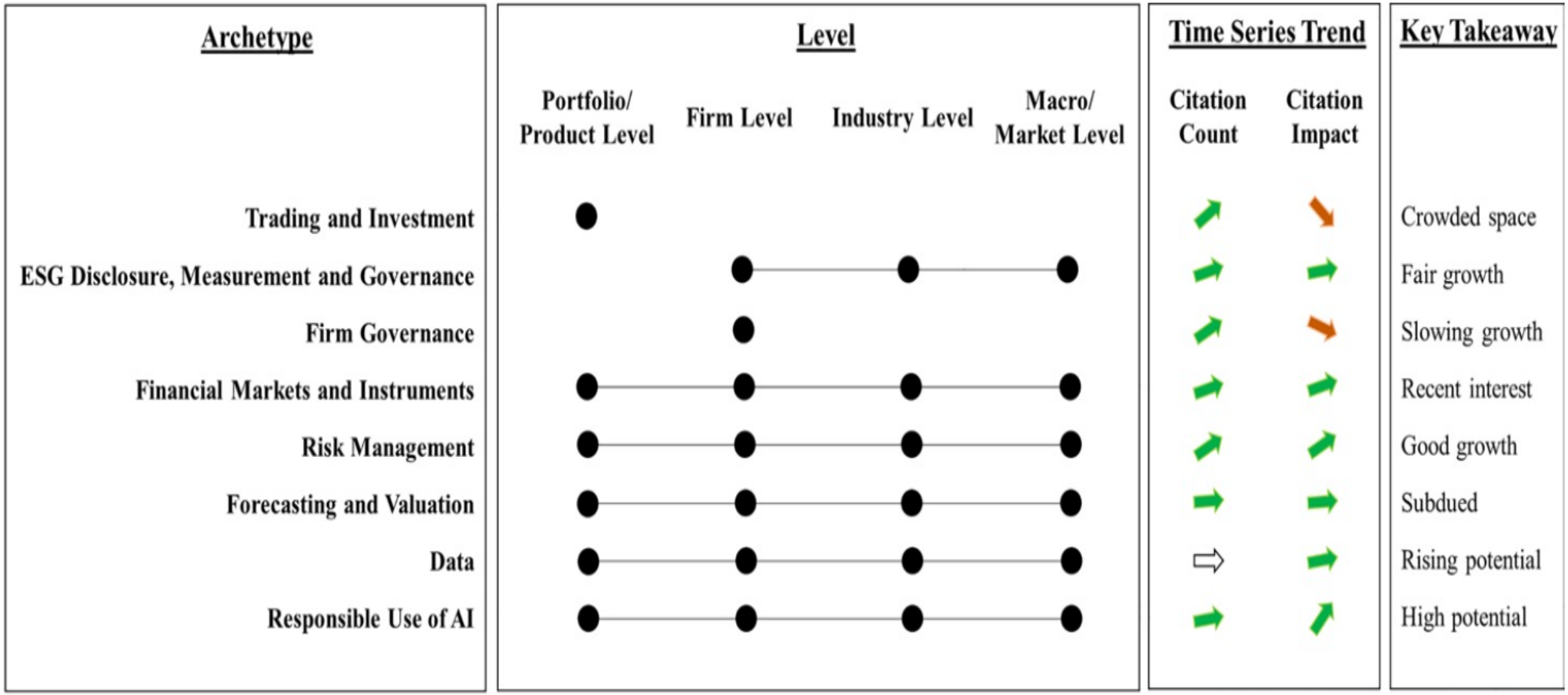
Source: Tristan Lim (2024). "Environmental, social, and governance (ESG) and artificial intelligence in finance: State-of-the-art and research takeaways." Artificial Intelligence Review 57, no. 4 (2024): 1-64.

Bump Graph Analysis of dominant AI Techniques



Source: Tristan Lim (2024). "Environmental, social, and governance (ESG) and artificial intelligence in finance: State-of-the-art and research takeaways." Artificial Intelligence Review 57, no. 4 (2024): 1-64.

Environmental, Social, and Governance (ESG) and Artificial Intelligence in Finance: Research Takeaways



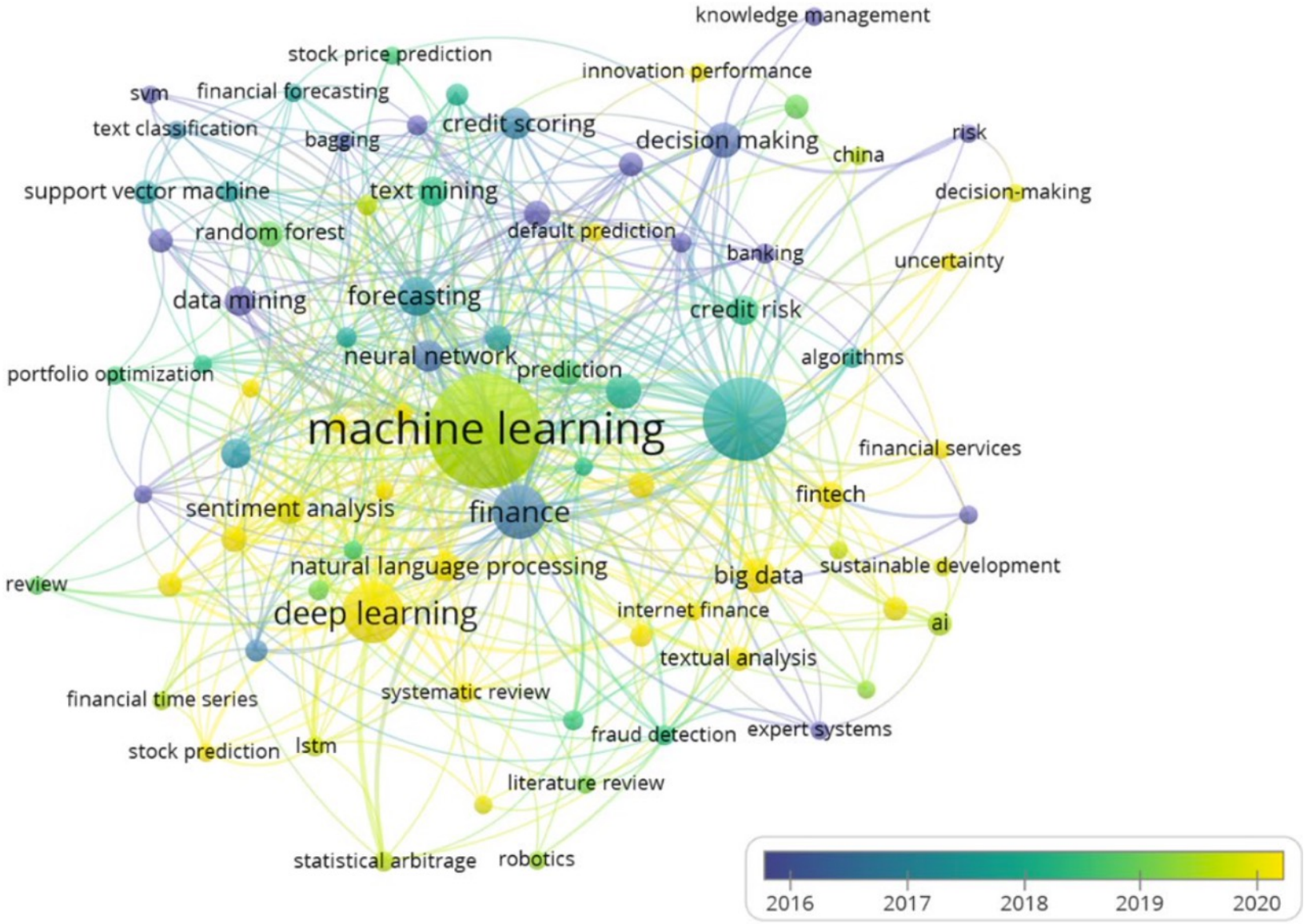
Source: Tristan Lim (2024). "Environmental, social, and governance (ESG) and artificial intelligence in finance: State-of-the-art and research takeaways." Artificial Intelligence Review 57, no. 4 (2024): 1-64.

Applications of DL in Financial Economics

- Deep learning for financial economics
- Deep learning for macroeconomics and monetary economics
- Deep learning for agricultural and natural resource economics
- Deep learning for industrial organization
- Deep learning for urban, rural, regional, real estate, and transportation economics
- Deep learning for health, education, and welfare
- Deep learning for business administration and microeconomics

Source: Zheng, Yuanhang, Zeshui Xu, and Anran Xiao. "Deep learning in economics: a systematic and critical review." Artificial Intelligence Review (2023): 1-43.

Machine Learning and Deep Learning in Finance

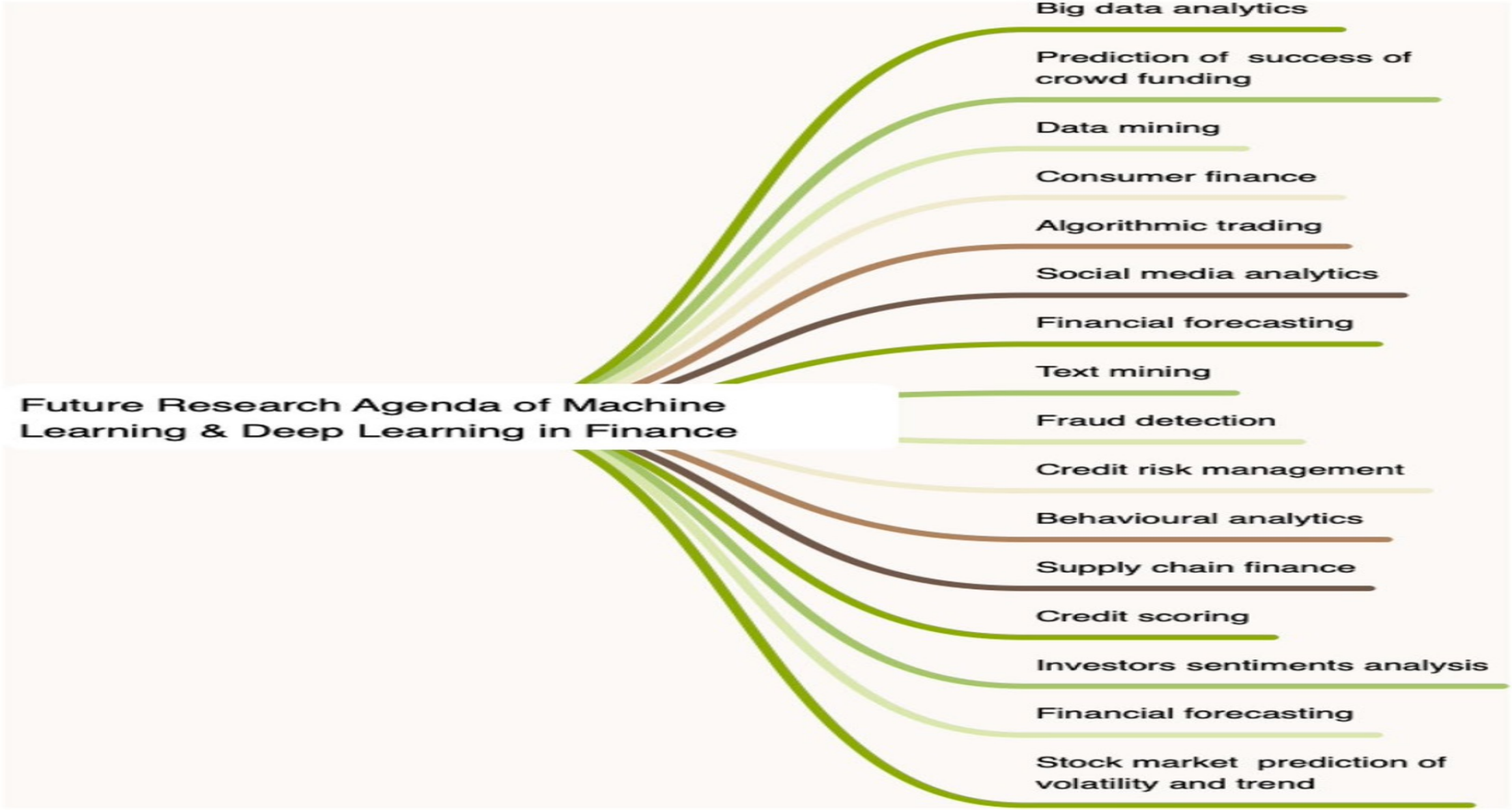


(2016-2022)

VOSviewer

Source: Biju, Ajitha Kumari Vijayappan Nair, Ann Susan Thomas, and J. Thasneem. "Examining the research taxonomy of artificial intelligence, deep learning & machine learning in the financial sphere—a bibliometric analysis." Quality & Quantity (2023): 1-30.

Machine Learning and Deep Learning in Finance

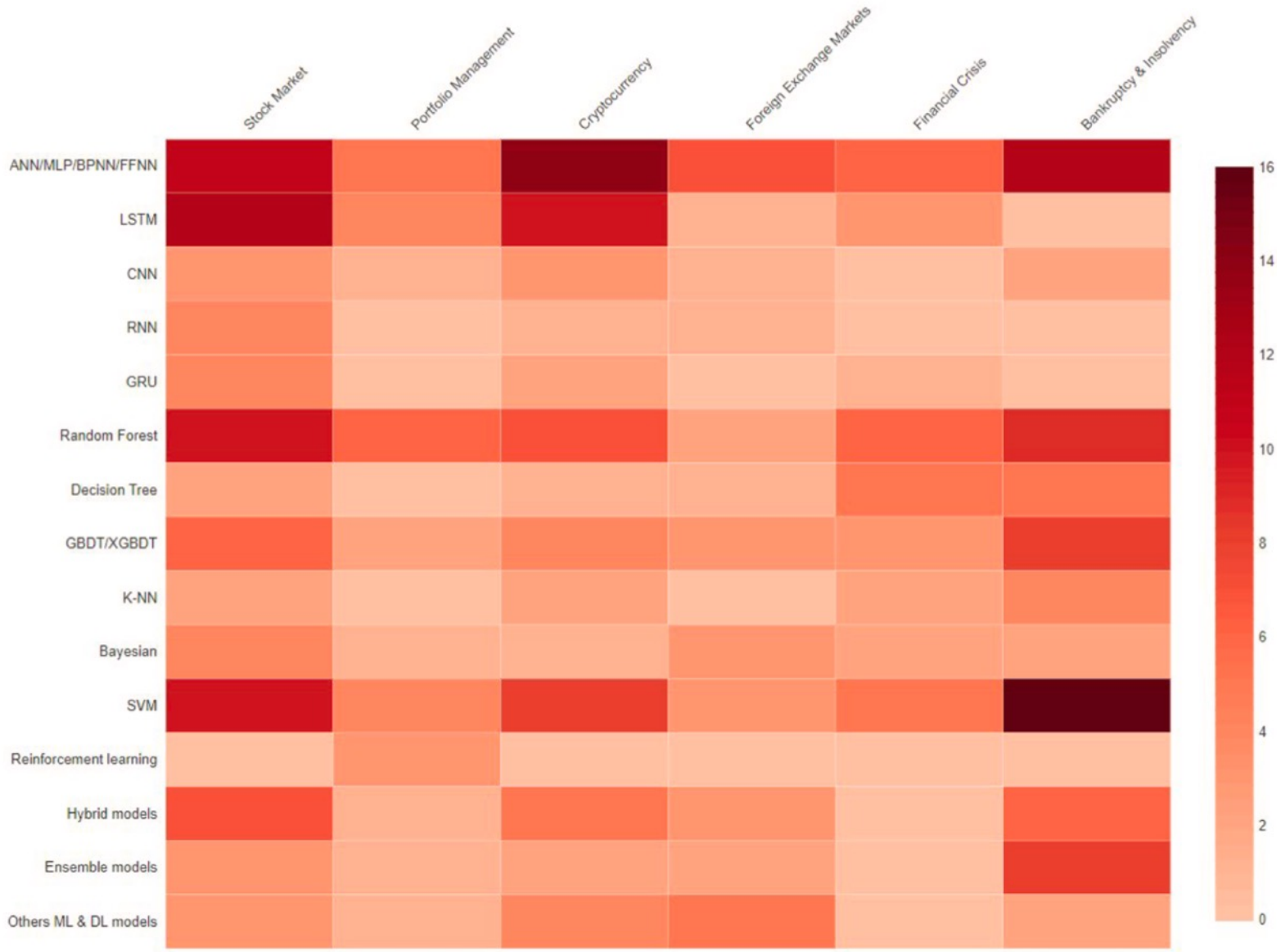


Source: Biju, Ajitha Kumari Vijayappan Nair, Ann Susan Thomas, and J. Thasneem. "Examining the research taxonomy of artificial intelligence, deep learning & machine learning in the financial sphere—a bibliometric analysis." Quality & Quantity (2023): 1-30.

Financial Applications of Machine Learning

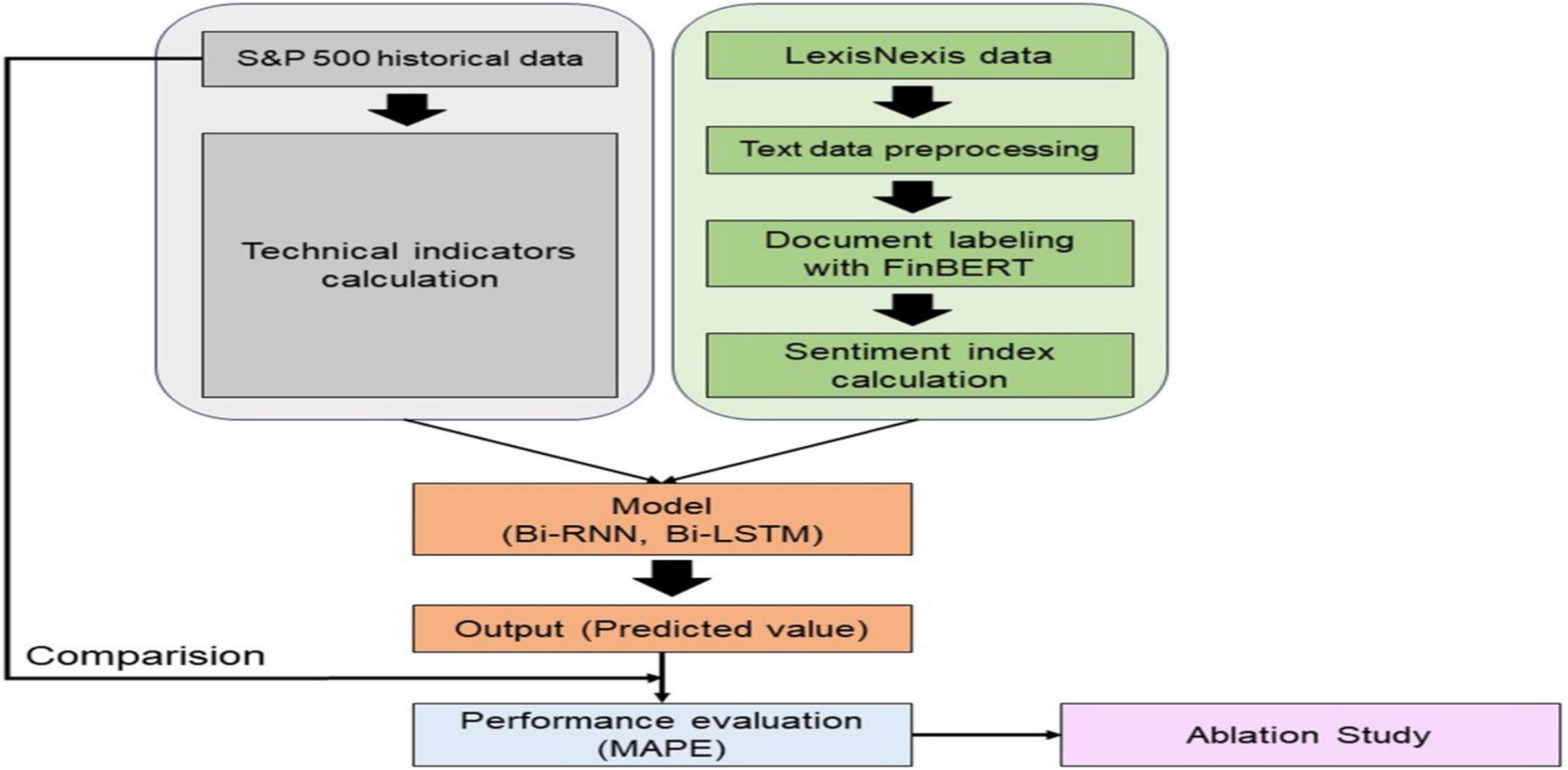
Application – Model Heatmap

Stock Market
Portfolio Management
Cryptocurrency
Foreign Exchange Market
Financial Crisis
Bankruptcy and Insolvency



Source: Nazareth, Noella, and Yeruva Yenkata Ramana Reddy. "Financial applications of machine learning: a literature review." Expert Systems with Applications (2023): 119640.

Deep-learning-based stock market prediction incorporating ESG sentiment and technical indicators



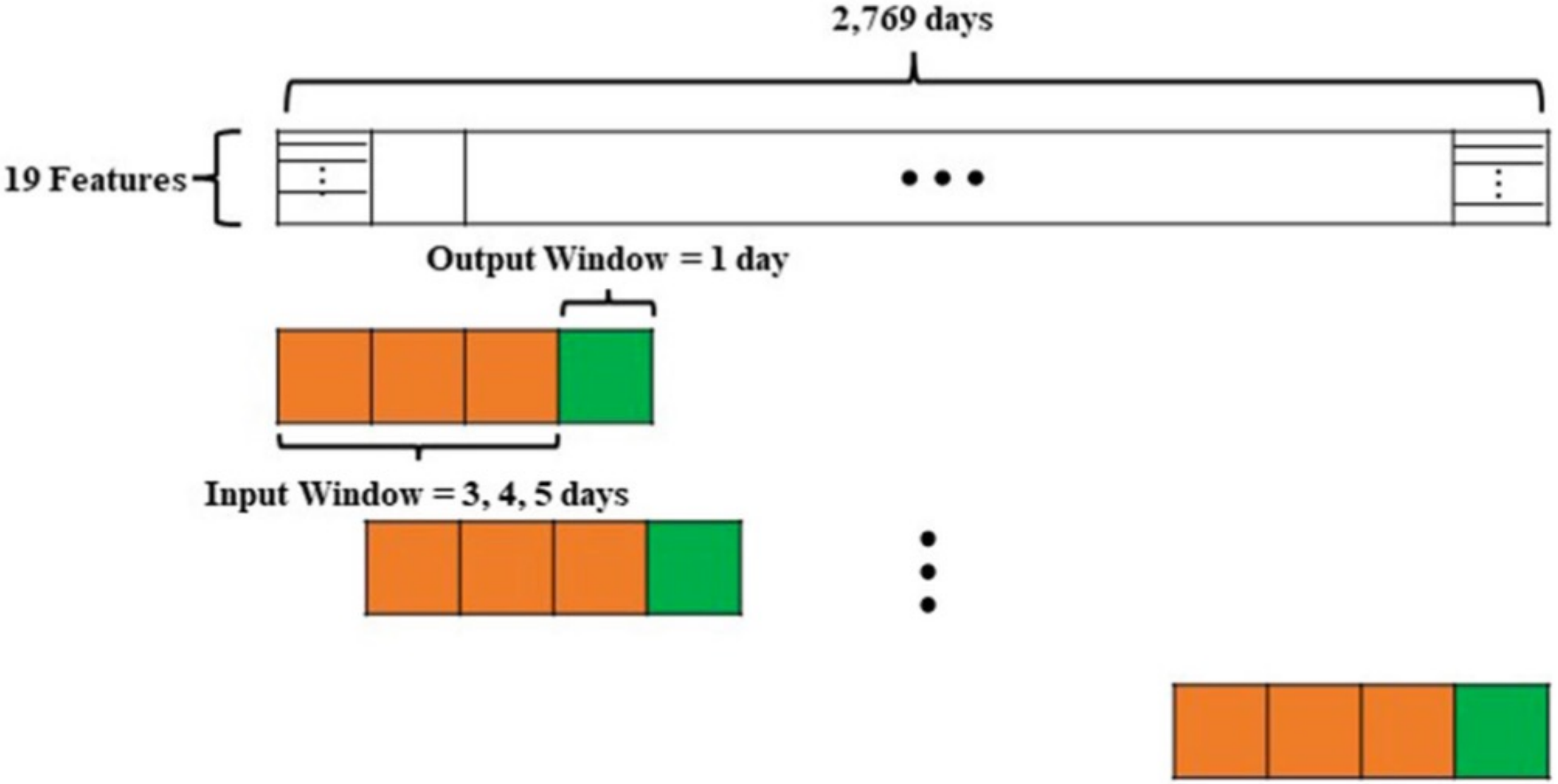
Source: Lee, Haein, Jang Hyun Kim, and Hae Sun Jung. "Deep-learning-based stock market prediction incorporating ESG sentiment and technical indicators." Scientific Reports 14, no. 1 (2024): 10262.

Technical indicators in stock market prediction

Indicator	Description
Simple Moving Average (SMA)	Provides a smoothing effect on price data over a designated time frame
Exponential Moving Average (EMA)	Provides a smoother perspective of price trends, emphasizing recent data
Chaikin Accumulation/Distribution Line (AD)	Measures cumulative buying and selling pressure for predicting price trends
Average Directional Movement Index (ADX)	Mean directional movement indicator
Kaufman Adaptive Moving Average (KAMA)	Adapts to changing market conditions, aiding in identifying optimal entry and exit points
Moving Average Convergence/Divergence (MACD)	Convergence and divergence of moving averages
Relative Strength Index (RSI)	Evaluates asset's overbought or oversold conditions, guiding potential reversals
Parabolic Stop and Reverse (PSAR)	Offers dynamic stop-loss levels, crucial for risk management
Momentum (MOM)	Measure the rate of change
Rate of Change (ROC)	Measure the percentage change in price from a previous period to the current period
Signal	Provide partial visual smoothing of technical indicators and detect trend reversals and crossovers
Stochastic RSI	Combination of the RSI and Stochastic indicator
Stochastic Oscillator	Relative position of prices over a given period

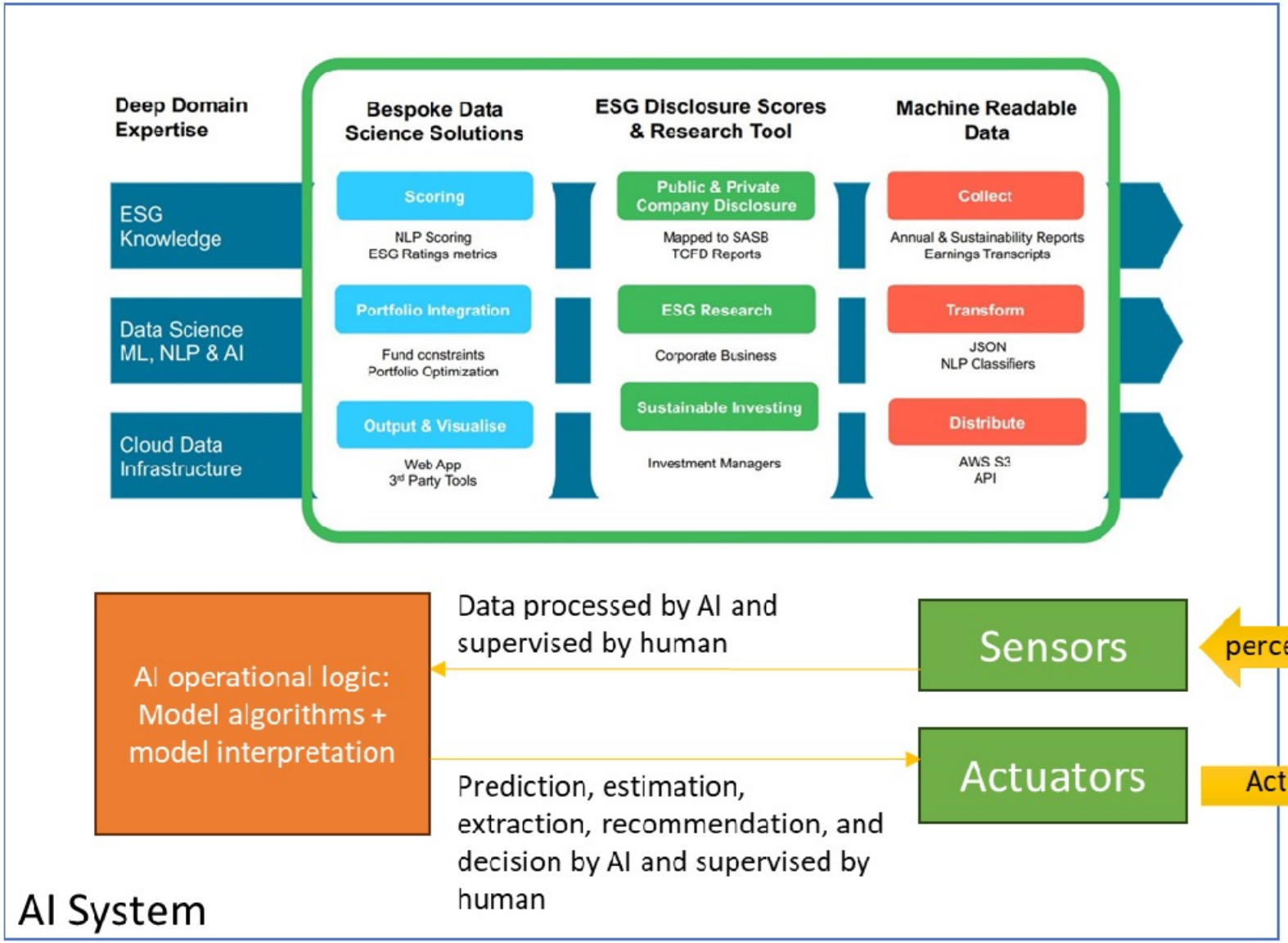
Source: Lee, Haein, Jang Hyun Kim, and Hae Sun Jung. "Deep-learning-based stock market prediction incorporating ESG sentiment and technical indicators." Scientific Reports 14, no. 1 (2024): 10262.

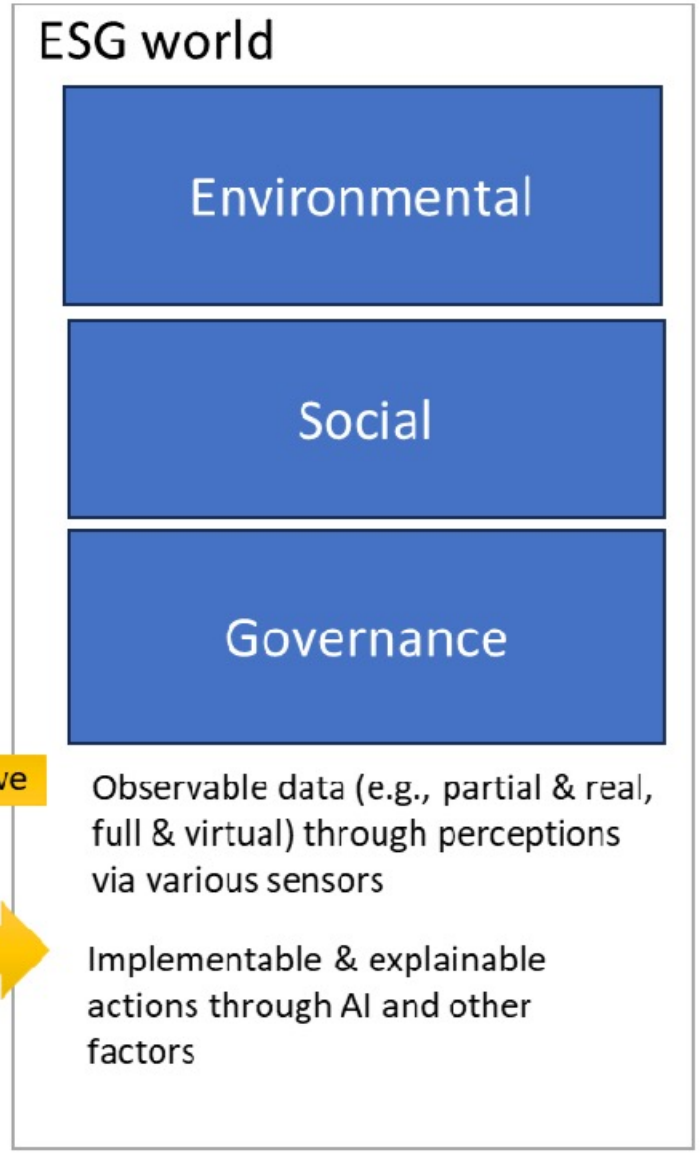
Technical indicators in stock market prediction



Source: Lee, Haein, Jang Hyun Kim, and Hae Sun Jung. "Deep-learning-based stock market prediction incorporating ESG sentiment and technical indicators." Scientific Reports 14, no. 1 (2024): 10262.

AI System and ESG World





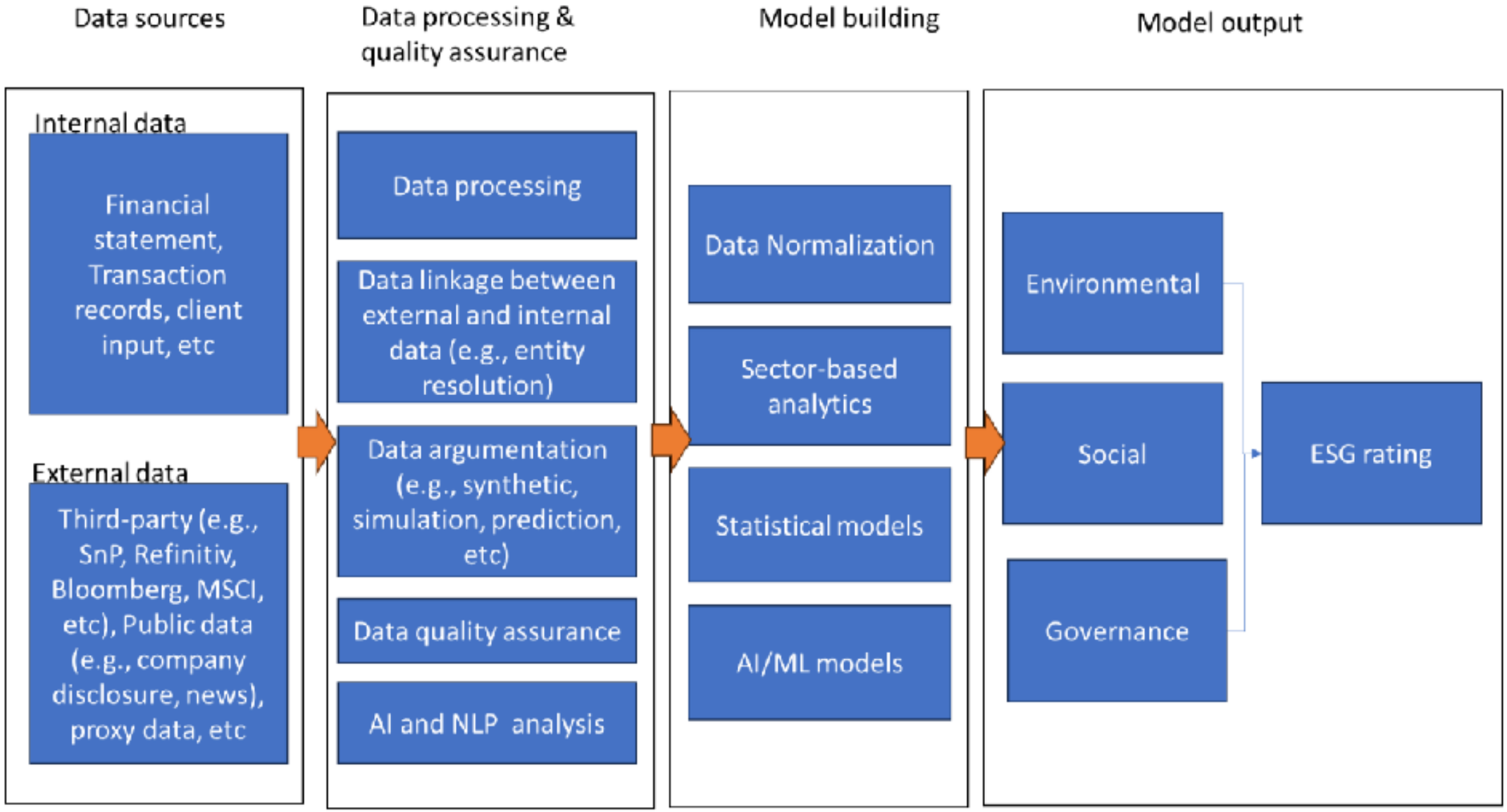
Source: Xu, Jun. (2024) "AI in ESG for Financial Institutions: An Industrial Survey." arXiv preprint arXiv:2403.05541 (2024).

AI Solution for Net Zero Model

Challenges	Description	Potential AI Solution
Data Complexity and Volume	Handling vast and complex data related to emissions across various industries.	Use of big data analytics and ML to process and analyze large datasets efficiently.
Dynamic Nature of Emissions	Emissions vary over time and by industry, requiring adaptable models. Different scopes may need different approaches.	Development of flexible AI models that can adapt to varying emission patterns and industry-specific factors.
Integration of Diverse Data Sources	Combining data from different sources and sectors for comprehensive analysis.	Implementing AI algorithms capable of integrating and synthesizing diverse data streams for holistic insights.
Predictive Accuracy	Accurately predicting future emissions and the effectiveness of mitigation strategies.	Employing predictive analytics and advanced ML techniques for precise forecasting of emissions and strategy outcomes.
Balancing Economic and Environmental Goals	Ensuring that net zero strategies are economically viable.	AI-driven optimization models that balance economic factors with environmental impact to identify cost-effective sustainability strategies.

Source: Xu, Jun. (2024) "AI in ESG for Financial Institutions: An Industrial Survey." arXiv preprint arXiv:2403.05541 (2024).

Data Analytics for ESG Ranking & Scoring



Source: Xu, Jun. (2024) "AI in ESG for Financial Institutions: An Industrial Survey." arXiv preprint arXiv:2403.05541 (2024).

Traditional and AI-Power ESG Rating

Aspect	Traditional ESG Rating Agencies	AI-Powered ESG Rating Organizations
Data Processing and Analysis	Rely on manual data collection and analysis, subject to human bias, e.g., analyst.	Use algorithms, ML, and NLP to analyze large data sets efficiently, reducing human bias.
Speed and Scalability	Slower due to manual processes; limited in the number of companies assessed. Usually, update semiannaully or annually	Faster updates and ability to scale up to assess more companies due to automation. Typically, update in daily or weekly.
Consistency and Objectivity	Potential for inconsistencies and subjective interpretations.	More consistent and objective, though not immune to underlying data biases.
Predictive Insights	Focus on current and past performance based on available data.	Capable of offering predictive insights about future ESG performance and risks.
Customization and Flexibility	Limited customization and flexibility due to manual processes.	Greater customization and flexibility, adapting quickly to new data sources or changing ESG criteria.

Source: Xu, Jun. (2024) "AI in ESG for Financial Institutions: An Industrial Survey." arXiv preprint arXiv:2403.05541 (2024).

ESG Data Providers

Provider	Description	Website
Bloomberg	Provide global coverage of ESG data; mainly on listed or public companies	https://fin-science.com/en/blog/esg/top-5-esg-data-providers-rating-and-report/
BvD (Moody's)	Provide ESG ratings, analytics, sustainability ratings, and sustainable finance reviewer/certifier services using data from Moody's. The BvD Orbis users can now add RepRisk's ESG risk metrics to the already substantial arsenal of company information available to you through our flagship global database of more than 200 million private companies	https://www.bvdinfo.com/en-gb/blog/compliance-and-financial-crime/product-update-using-orbis-to-assess-environmental-social-and-governance-esg-risk
CDP	CDP (formerly the Carbon Disclosure Project) ESG Rating is a unique rating that identifies the best ESG-integrated investment funds, based on their ESG performance	https://www.cdp.net/en/scores/cdp-scores-explained
FTSE Russell / LSEG	Provide ESG ratings using a company's Theme Exposure and Theme level score assessment to calculate range of assessments that allow investors to understand a company's ESG practices in multiple dimensions.	https://www.ftserussell.com/products/indices/esg
ISS	Provide ESG data and research on companies. The system's goal is to take the ESGY Scorecard directly to the industry level.	https://www.issgovernance.com/esg/

Source: Xu, Jun. (2024) "AI in ESG for Financial Institutions: An Industrial Survey." arXiv preprint arXiv:2403.05541 (2024).

ESG Data Providers

Provider	Description	Website
MSCI	Offer data on ESG. Its ratings aim to measure a company's management of financially relevant ESG risks and opportunities.	https://www.msci.com/our-solutions/esg-investing/esg-ratings
Refinitiv / LSEG	Provides comprehensive ESG data. Their ESG scores are designed to transparently and objectively measure a company's relative ESG performance, commitment and effectiveness across 10 main themes (emissions, environmental product innovation, diversity and inclusion, human rights, shareholders, etc.) based on publicly-reported data	https://www.refinitiv.com/content/dam/marketing/en_us/documents/methodology/refinitiv-esg-scores-methodology.pdf
Sustainalytics	An independent ESG research and rating agency that provides data and research on the environmental and social performance of companies.	https://connect.sustainalytics.com/hubfs/SFS/Sustainalytics%20ESG%20Risk%20Rating%20-%20FAQs%20for%20Corporations.pdf
S&P Global	Provide access to transparently disclosed ESG data points for companies assessed in the S&P	https://www.spglobal.com/esg/solutions/esg-data-intelligence

Source: Xu, Jun. (2024) "AI in ESG for Financial Institutions: An Industrial Survey." arXiv preprint arXiv:2403.05541 (2024).

Machine Learning in Stock Market Prediction

Authors	Index/Stock, Country	Model	Time Period	Performance evaluation	Task
Seong & Nam (2022)	SPX index, KOSPI index, US and South Korea	Attention based CNN + LSTM	1st January 1971 to 31st December 2019	Accuracy for short term = 0.602 Medium term = 0.660 and long term = 0.688 and Sharpe Ratio for short = 5.490 medium = 2.895 and long term = 3.239	Classification and Trading
Akhtar, Zamani, Khan, Shatat, & Dilshad, (2022)	BSE 500, DJIA, NASDAQ, India and US	Random Forest, SVM and LSTM	Extracted from Kaggle	Accuracy = 80.3 %	Forecasting
Gupta, Bhattacharjee, & Bishnu (2022)	CNX-Nifty, India	GRU based Stock-Net model	1st April 1996 to 6th January 2020	RSME = 0.0896 MAE = 69.9396 MAPE = 0.8203	Forecasting
Ma & Yan (2022)	CSI 300 index and Shanghai stock index, China	CNN	January 2007 to December 2021	Accuracy = 69.39 %	Forecasting
Park, Kim, & Kim (2022)	S&P500, SSE, and KOSPI200, US, China and South Korea	LSTM_Random Forest (Hybrid)	01st Aug 2002 to 13th September 2018	Regression: RMSE = 1.89, MAE = 1.41, and MAPE = 0.51 Classification: Accuracy = 59.83 Balanced accuracy = 59.95	Classification, forecasting and trading
Bhandari, et al. (2022)	S&P 500 index, US	LSTM	2006 to 2020	RMSE = 40.4574, MAPE = 0.7989 and R = 0.9976	Forecasting
Banik, Sharma, Mangla, Mohanty, & Shitharth (2022)	ICICI Bank and NIFTY-Bank from National Stock Exchange of India (Kaggle dataset)	LSTM	1st Jan 2020 to 31st July 2020	RMSE = 4.13 %, MAE = 3.24 %, and MAPE = 1.21 %	Forecasting
Pokhrel, et al. (2022)	Nepal Stock Exchange Limited (NEPSE)	LSTM, GRU, and CNN	17th July 2016 to 15th January 2020	RMSE = 10.4660 ± 0.6836, MAPE = 0.6488 ± 0.0502 and R = 0.9874 ± 0.0009	Forecasting

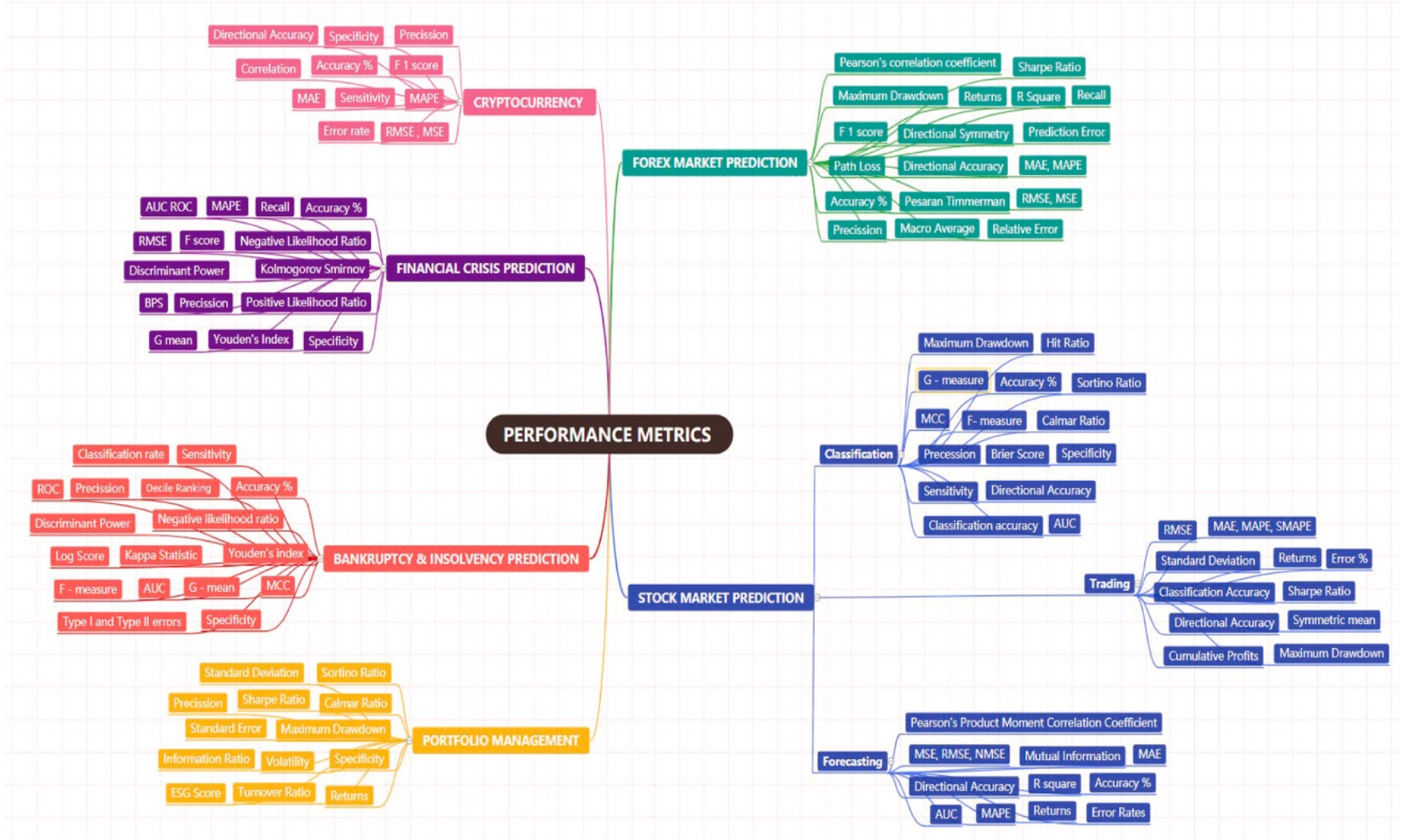
Source: Nazareth, Noella, and Yeruva Yenkata Ramana Reddy. "Financial applications of machine learning: a literature review." Expert Systems with Applications (2023): 119640.

Machine Learning in Portfolio Management

Title	Index/stocks	Models	Portfolio strategy	Time period	Transaction cost	Number of stocks in portfolio	Performance evaluation
Betancourt & Chen (2021)	Bitcoin, Ethereum, Litecoin and others. the number of active assets that can be exchanged for USDT incremented from three to 85	Deep Reinforcement Learning	Agent trader	17th August 2017 to 01st November 2019	With transaction fees (except BNB) 0.1 % and fees BNB 0.05 %	Dynamic number of assets	Average daily returns of over 24 %, Sharpe ratio = 0.46
Juan (2022)	20 stocks from CSI 300 and 20 stocks from the S&P 500	SVM, Random Forest, and Attention-based LSTM	Mean-Variance Portfolio compared to 1/N	May 4, 2012 to August 4, 2020.	Without transaction cost	Less than 20 stocks	Accuracy = 92.59 % (CSI 300) and 88.52 % (S&P 500), Sharpe ratio of 9.31 (CSI 300) and 2.77 (S&P 500)
Pinelis and Ruppert (2022)	NYSE, AMEX, and NASDAQ indices	Random Forest, Elastic Net and Linear model	Buy and Hold market strategy	1927–2019	With an increase in transaction cost @ 1bps, 10bps and 14bps for trading approximately 1 % of daily volume	N/A	Sharpe Ratio = 0.68, Annualized returns = 12 %-13 % and standard deviation = 13 to 15 %
Barua & Sharma (2022)	MSCI Asia Pacific sector indices: Energy, Utilities Consumer, Health Care, Communication Services, Information Technology, Consumer Staples, Discretionary, Materials, Financials & Industrials	Hybrid: CNN-BiLSTM	Buy and Hold market strategy	01st April 2002 to 31st March 2022	With transaction costs @ 25 basis points	N/A	Sharpe Ratio = 2.55 and Herfindahl Index = 0.128

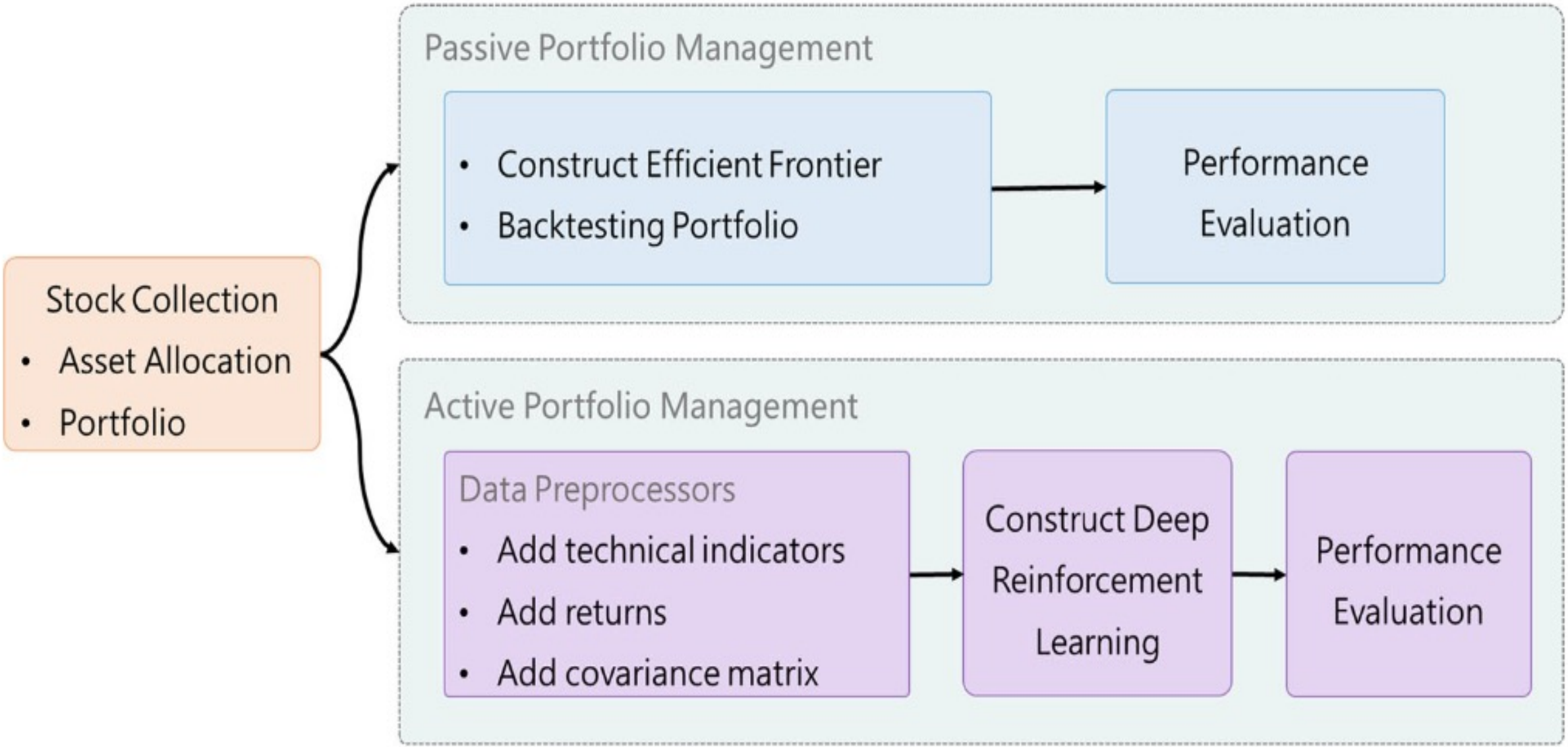
Source: Nazareth, Noella, and Yeruva Yenkata Ramana Reddy. "Financial applications of machine learning: a literature review." Expert Systems with Applications (2023): 119640.

Performance Metrics of Financial Applications of ML



Source: Nazareth, Noella, and Yeruva Yenkata Ramana Reddy. "Financial applications of machine learning: a literature review." Expert Systems with Applications (2023): 119640.

Deep Reinforcement Learning in Portfolio Management



Source: Min-Yuh Day, Ching-Ying Yang, and Yensen Ni (2024), "Portfolio Dynamic Trading Strategies Using Deep Reinforcement Learning", Soft Computing, Volume 28, August 2024, pp. 8715–8730.

Deep Reinforcement Learning in Portfolio Management

References	Data set	Data period	Features setting	Deep learning method	Performance criteria	Transaction cost
Huang et al. (2020)	CSI300 index, 3 stocks in CSI500	2007–2020	OCHLV	DDPG	Average return: 0.000949 Annual Sharpe: 1.707 Annual Sortino: 2.946 MDD: 8.09%	Yes, 0.0025
Zhang et al. (2020)	50 futures contracts	2005–2019	OCHLV, MA, MACD, RSI	LSTM, DQN, PG, A2C	$E(R)$: 1.258 Std.(R): 0.976 Sharpe: 1.288, Sortino: 2.220 MDD: 0.002	Yes, 0.0001
Rundo (2019)	EUR/USD	2004–2018	OCHLV, Technical indicators	LSTM, DQN	ROI Max = 98.81% MD Max = 19.28%	No
Liu et al. (2021)	Bitcoin	2017–2020	OCHLV, technical indicators	LSTM, PPO	Profits rate = 1.46%	Yes, 0.0025
Darapaneni et al. (2020)	16 index from Indian market	2011–2019	OCHLV, SMA	Q-learning	Annualized return = 8.54% Max drawdown = 1.44%	Yes, 0.0030
Wu et al. (2020)	3 stocks from US, 3 stocks from UK, and 3 stocks from China	2008–2018	OCHLV, MA, EMA, ACD, BIAS	GRU, DQN, DPG	Return = 215.9% SR = 1.9%	No
This study **Day et al. (2023)**	227 constituent stocks from MSCI US ESG Select Index	2000–2020	OCHLV, technical indicators	Proposed DRLPMESG framework, PPO	Annual return = 45.58% SR = 1.37	No

Source: Min-Yuh Day, Ching-Ying Yang, and Yensen Ni (2023), "Portfolio Dynamic Trading Strategies Using Deep Reinforcement Learning", Soft Computing, 30 July 2023, pp. 1-16.

Deep learning for Financial Economics

Appli-cation subfield	Article	Aim of study	Data set	Date size	Time span	Used models
Financial market	(Heaton et al., 2017)	Predict and classify financial market	Component stocks of the biotechnology IBB index	Weekly returns data	2012–2016	Stacked AE
	(Mishev et al., 2020)	Analyze sentiment in finance	Financial Phrase-Bank dataset, SemEval-2017 task dataset	4,845 English sentences, 2,510 news headlines	-	RNN, RNN, Attention, CNN, Dense Network
Stock market	(Zhong & Enke, 2019)	Forecast daily stock return	SPDR S&P 500 ETF (ticker symbol: SPY)	60 factors over 2,518 trading days	2003–2013	DNN, ANN
	(Chatzis et al., 2018)	Forecast crisis events	FRED and the SNL	More than 5,000 records	1996–2017	MXNET, DNN
	(Nikou et al., 2019)	Predict stock price	iShares MSCI United King-dom exchange	869 data	2015–2018	LSTM
	(Sharaf et al., 2021)	Predict stock price	Quandl dataset	-	2000–2019	LSTM, CNN, Stacked-LSTM, Bi-LSTM

Source: Zheng, Yuanhang, Zeshui Xu, and Anran Xiao. "Deep learning in economics: a systematic and critical review." Artificial Intelligence Review (2023): 1-43.

Journal Articles of ML and AI in Finance

Publication outlet	Number of articles	TC
Expert systems with applications	34	1174
Sustainability	34	169
IEEE access	28	159
Computational economics	22	105
European journal of operational research	16	842
Quantitative finance	12	80
Journal of behavioral and experimental finance	11	88
Journal of forecasting	10	115
Technological forecasting and social change	10	52
Applied soft computing	9	467

Source: Biju, Ajitha Kumari Vijayappan Nair, Ann Susan Thomas, and J. Thasneem. "Examining the research taxonomy of artificial intelligence, deep learning & machine learning in the financial sphere—a bibliometric analysis." Quality & Quantity (2023): 1-30.

Journal Articles of ML and AI in Finance

Publication outlet	Number of articles	TC
Decision support systems	9	351
Mathematics	8	48
Economic computation and economic cybernetics studies and research	7	19
Journal of enterprise information management	7	37
Journal of intelligent & fuzzy systems	7	23
Complexity	6	23
Financial innovation	6	28
Journal of international financial markets institutions & money	6	25
Knowledge-based systems	6	132
Technological and economic development of economy	6	182

Source: Biju, Ajitha Kumari Vijayappan Nair, Ann Susan Thomas, and J. Thasneem. "Examining the research taxonomy of artificial intelligence, deep learning & machine learning in the financial sphere—a bibliometric analysis." Quality & Quantity (2023): 1-30.

Deep learning: CNN, RNN

	Features	Advantages	Disadvantages	Main variants
CNN	a. Effectively reduce the dimension of large data images to small data (without affecting the results); b. Can retain the characteristics of the picture, similar to the principle of human vision.	a. The weight sharing strategy reduces the parameters that need to be trained, making the generalization ability of the trained model stronger; b. Pooling operation can reduce the spatial resolution of the network, so that the translation invariance of the input data is not required.	Depth models are prone to gradient dissipation.	GoogleNet, VGG, Deep Residual Learning
RNN	a. Long-term information can be effectively retained; b. Select important information to keep, and select "forget" for less important information.	The model is a depth model in time dimension, which can model the sequence content.	a. There are many parameters that need to be trained, which are prone to gradient dissipation or gradient explosion; b. Without feature learning ability.	LSTM, GRU

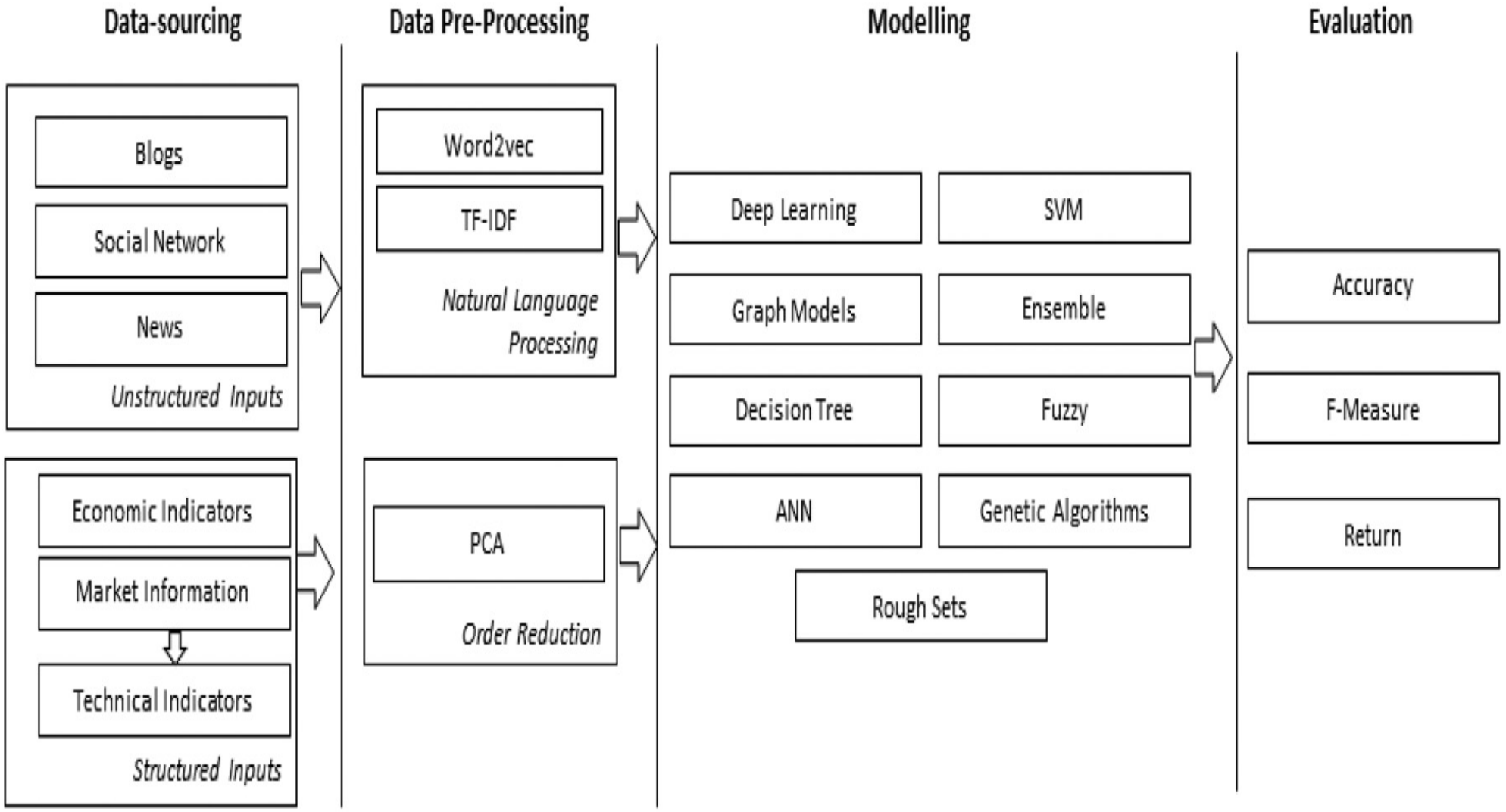
Source: Zheng, Yuanhang, Zeshui Xu, and Anran Xiao. "Deep learning in economics: a systematic and critical review." Artificial Intelligence Review (2023): 1-43.

Deep learning: AE, Transformer, DRL

	Features	Advantages	Disadvantages	Main variants
AE	a. Data dependent; b. Learn automatically from data samples.	a. Strong generalization; b. Can be used for dimension reduction; c. Can be used for feature detectors; d. Can be used for generation model.	a. Information is somewhat lost in the process of encoding and decoding; b. The compression ability only applies to samples similar to training samples.	Denoising AE, Stack AE, Undercomplete AE, Regular AE
Transformer	a. Self-attention mechanism; b. Focus on global information.	a. Enables to model more long-distance dependencies; b. Parallel computing.	a. High program complexity; b. Not Turing complete; c. Compute resource input average.	Linear Transformer, Sparse Transformer, Reformer, Set Transformer, Transformer-XL
DRL	a. Combine deep learning with reinforcement learning; b. End-to-End training.	a. Learn control strategies directly from high-dimensional raw data; b. Large numbers of samples can be produced for supervised study.	a. Difficult to achieve continuous motion control; b. Overestimation, that is, the estimated value function is larger than the true value function.	QR-DQN, Rainbow DQN

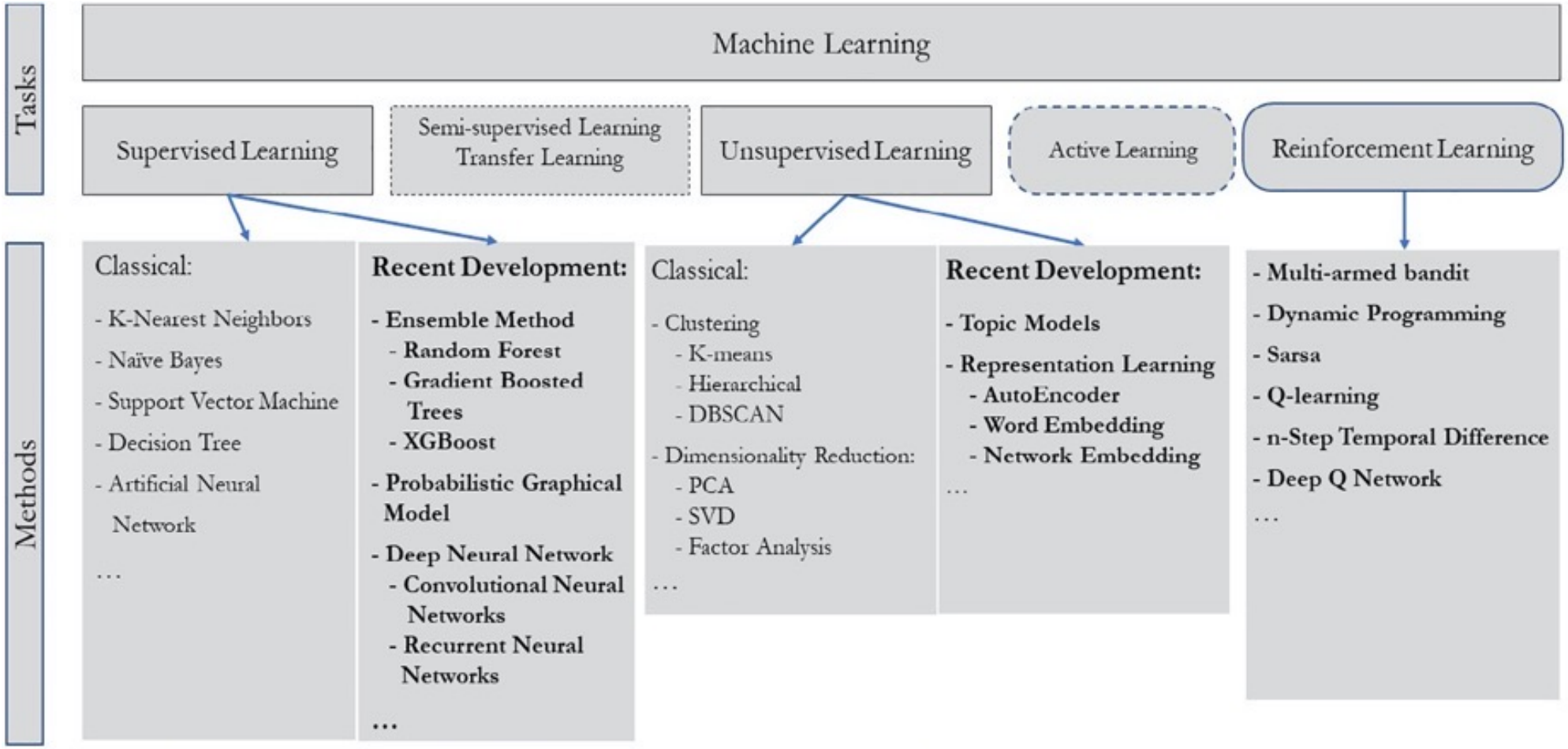
Source: Zheng, Yuanhang, Zeshui Xu, and Anran Xiao. "Deep learning in economics: a systematic and critical review." Artificial Intelligence Review (2023): 1-43.

Stock Market Movement Forecast:
ML Phases of the stock market modeling



Source: O. Bustos and A. Pomares-Quimbaya (2020), "Stock Market Movement Forecast: A Systematic Review." Expert Systems with Applications (2020): 113464.

Machine Learning Tasks and Methods



Note: Several entries in the diagram, e.g. word embedding or multi-armed bandit, refer to specific problem formulations for which a collection of methods exist.

: Tasks that take input data as given : Tasks that involve interactive data acquisition Dashed border: methods not elaborated in paper text

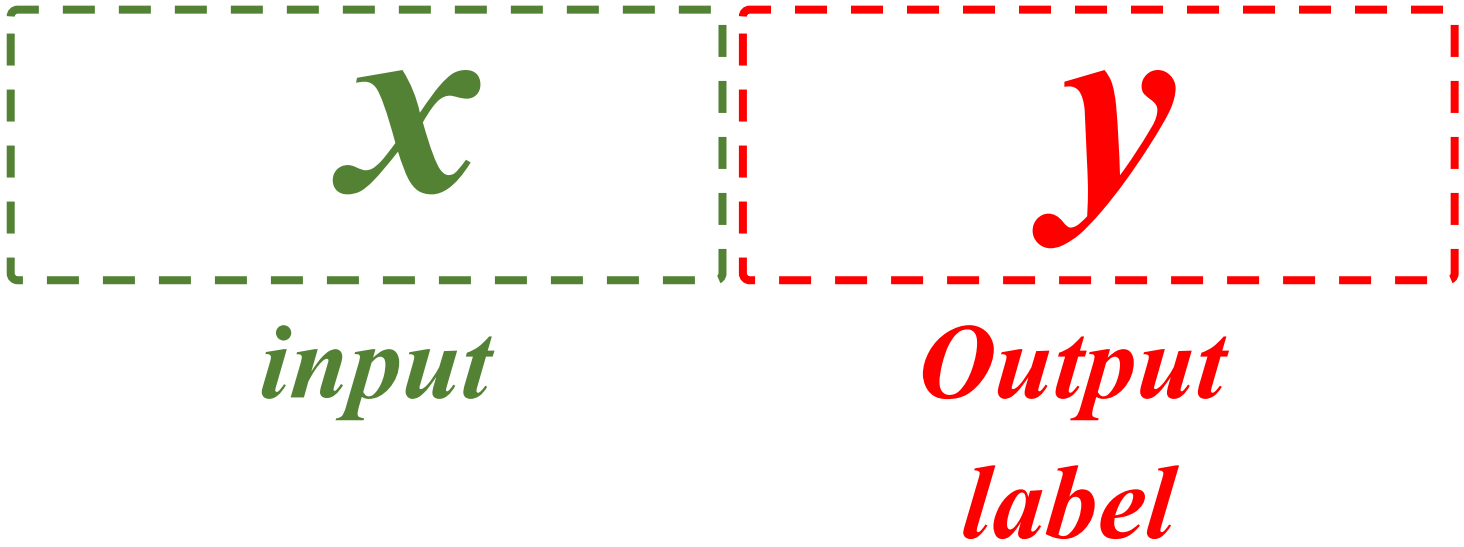
Bold type: highlights recent developments

Source: Liye Ma and Baohong Sun (2020), "Machine learning and AI in marketing – Connecting computing power to human insights." International Journal of Research in Marketing, 37, no. 3, 481-504.

Machine Learning

Machine Learning
Supervised Learning (Classification)
Learning from Examples

$$y = f(x)$$



Machine Learning
Supervised Learning (Classification)
Learning from Examples

$y = f(x)$

Example

```
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
7.0,3.2,4.7,1.4,Iris-versicolor
6.4,3.2,4.5,1.5,Iris-versicolor
6.9,3.1,4.9,1.5,Iris-versicolor
6.3,3.3,6.0,2.5,Iris-virginica
5.8,2.7,5.1,1.9,Iris-virginica
7.1,3.0,5.9,2.1,Iris-virginica
```

x

y

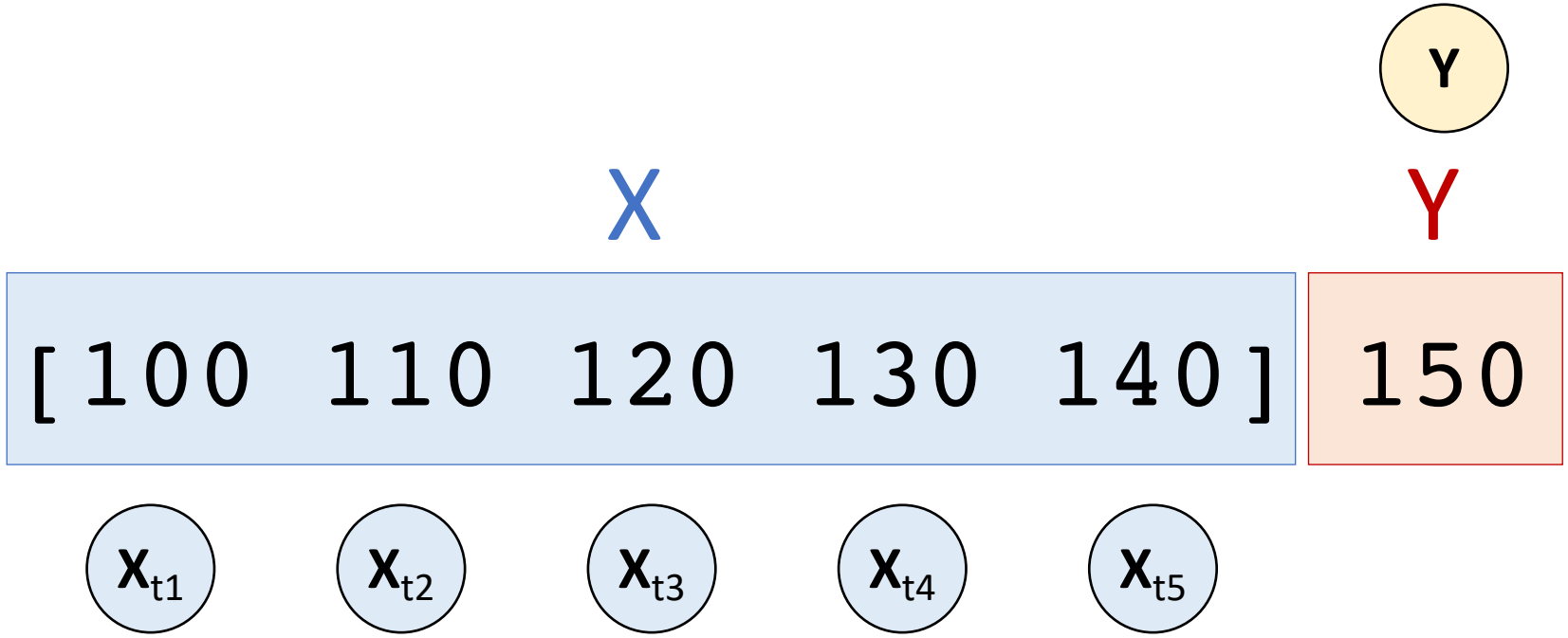
Time Series Data

[10, 20, 30, 40, 50, 60, 70, 80, 90]

X	Y
[10 20 30]	40
[20 30 40]	50
[30 40 50]	60
[40 50 60]	70
[50 60 70]	80
[60 70 80]	90

Time Series Data

[100, 110, 120, 130, 140, 150]



Linear function

$$y = f(x)$$

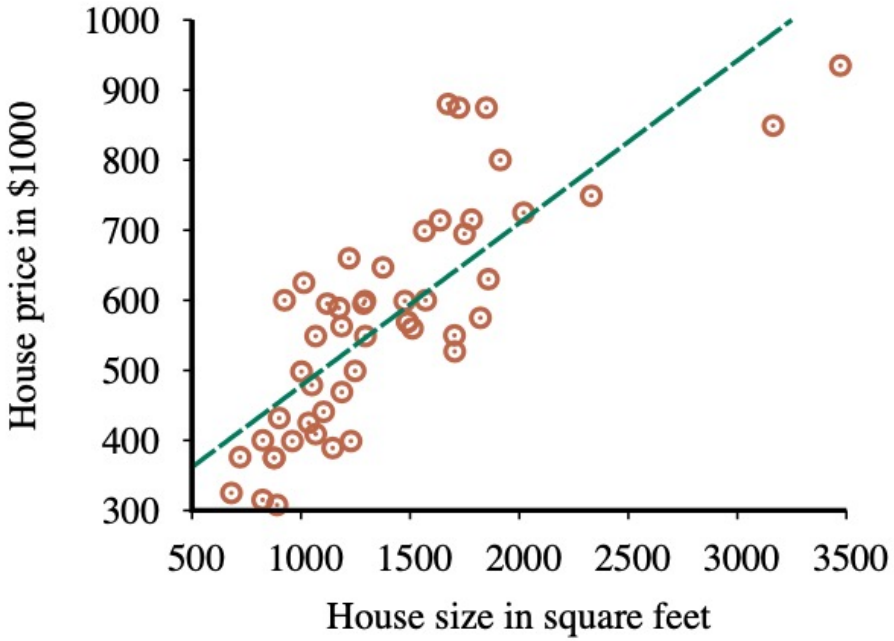
$$y = w_1 x + w_0$$

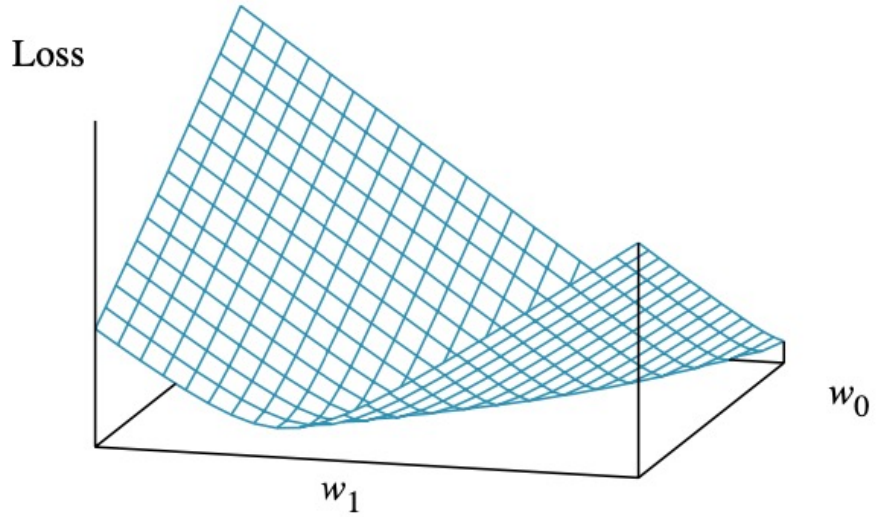
$$h_w(x) = w_1 x + w_0$$

Linear Regression Weight Space

$$h_w(x) = w_1 x + w_0$$

$$w^* = \text{argmin}_w \, Loss(h_w)$$





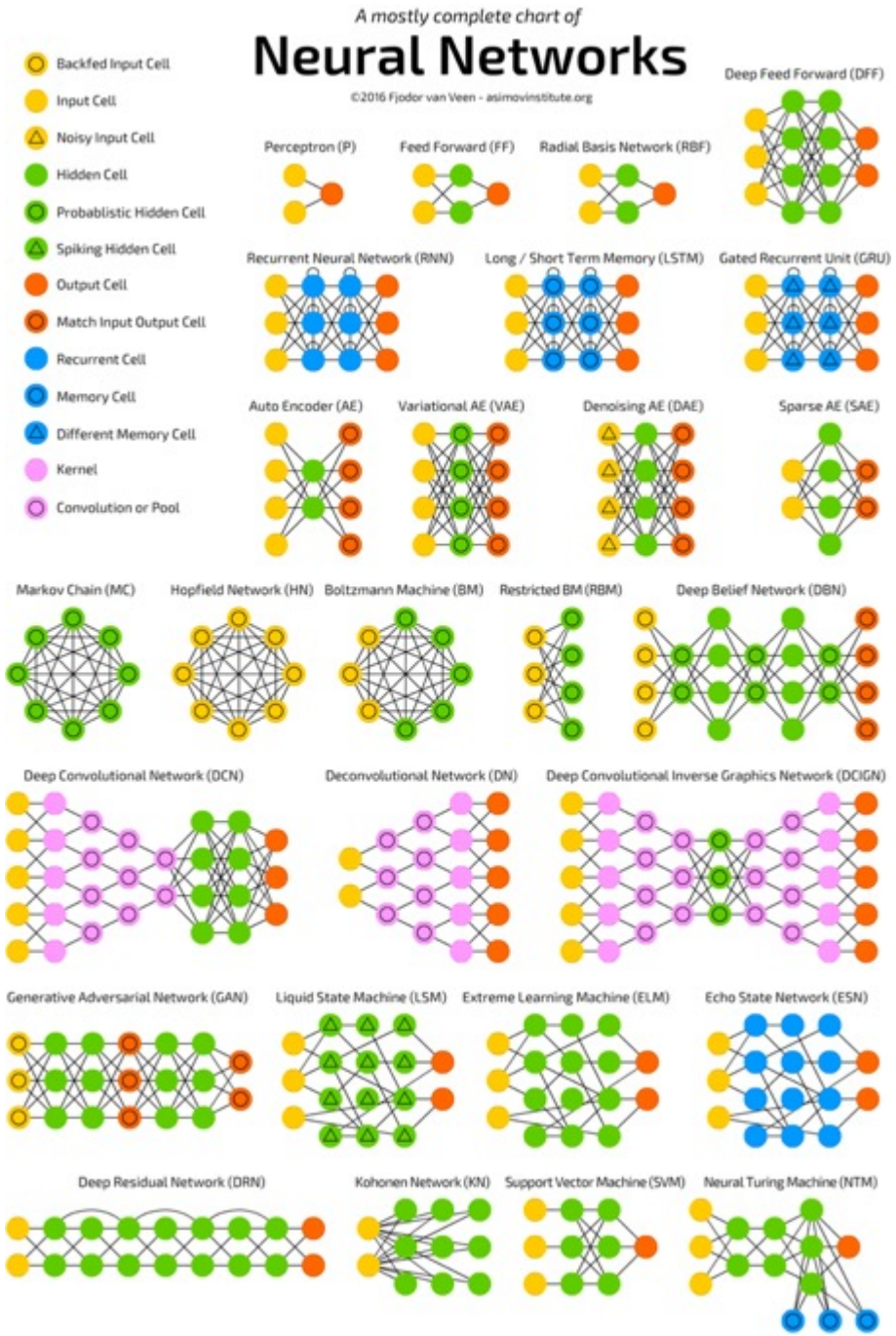
$y = 0.232\,x + 246$

Loss function for Weights (w_1, w_0)

Deep Learning

Deep Learning
and
Deep Neural Networks

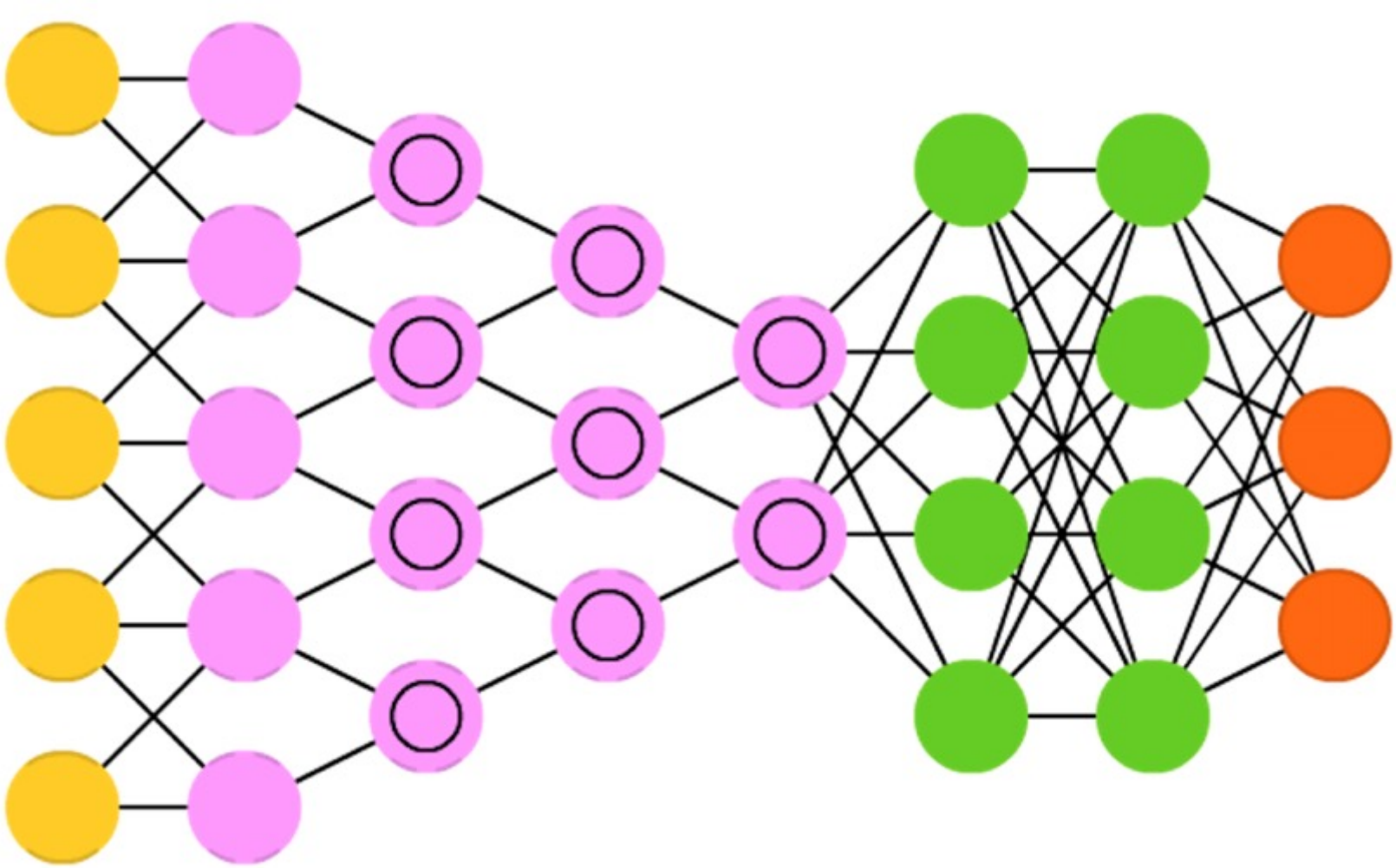
Neural Networks (NN)



Source: http://www.asimovinstitute.org/neural-network-zoo/

Convolutional Neural Networks

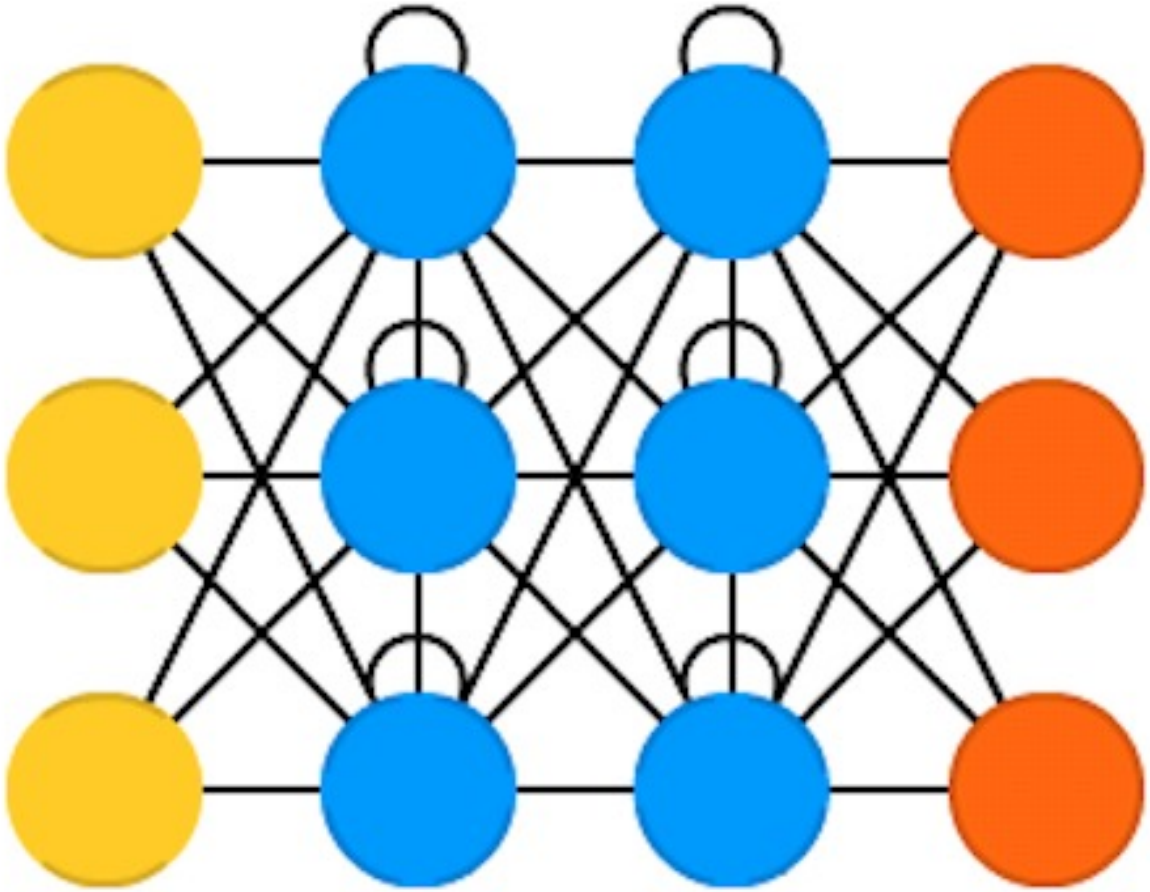
(CNN or Deep Convolutional Neural Networks, DCNN)



LeCun, Yann, et al. “Gradient-based learning applied to document recognition.” Proceedings of the IEEE 86.11 (1998): 2278-2324.

Source: http://www.asimovinstitute.org/neural-network-zoo/

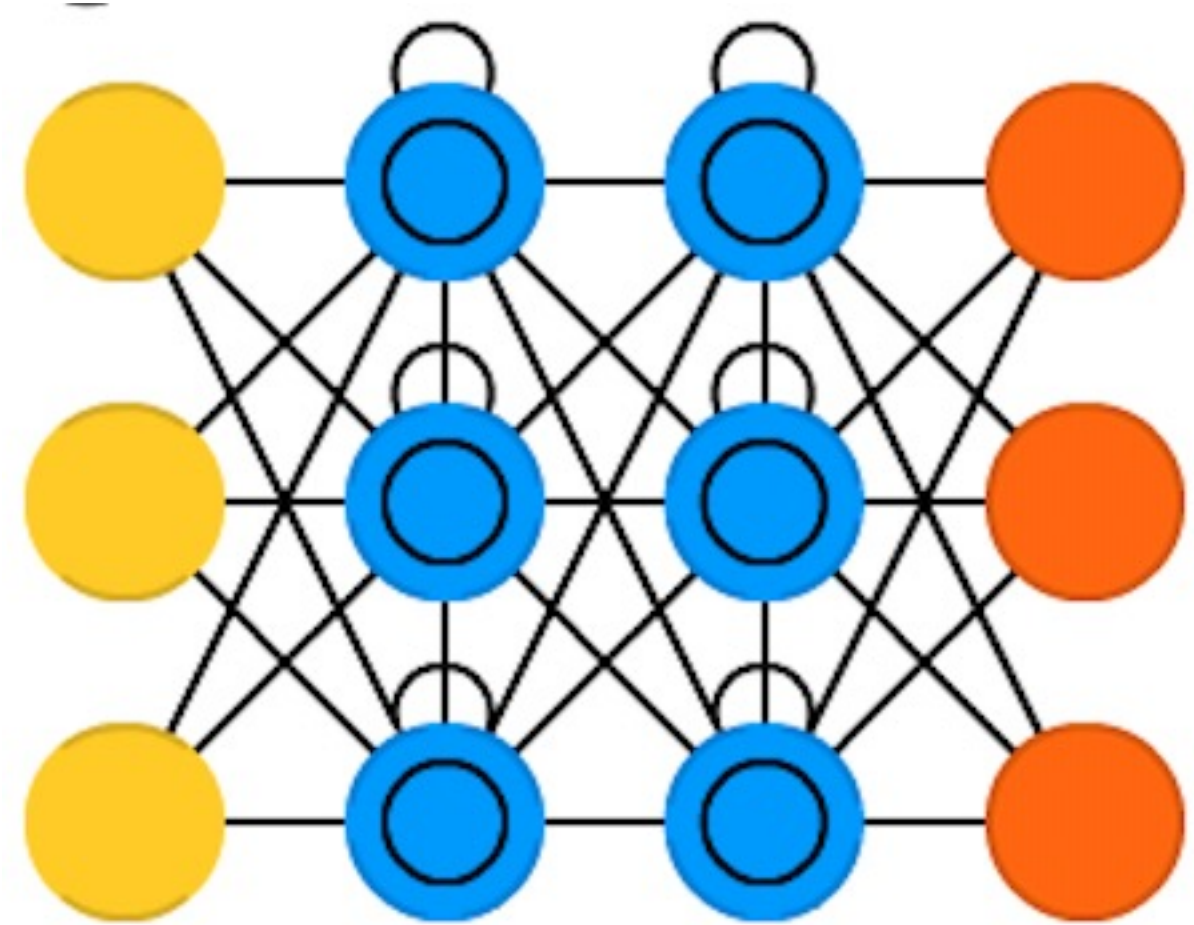
Recurrent Neural Networks (RNN)



Elman, Jeffrey L. “Finding structure in time.” Cognitive science 14.2 (1990): 179-211

Source: http://www.asimovinstitute.org/neural-network-zoo/

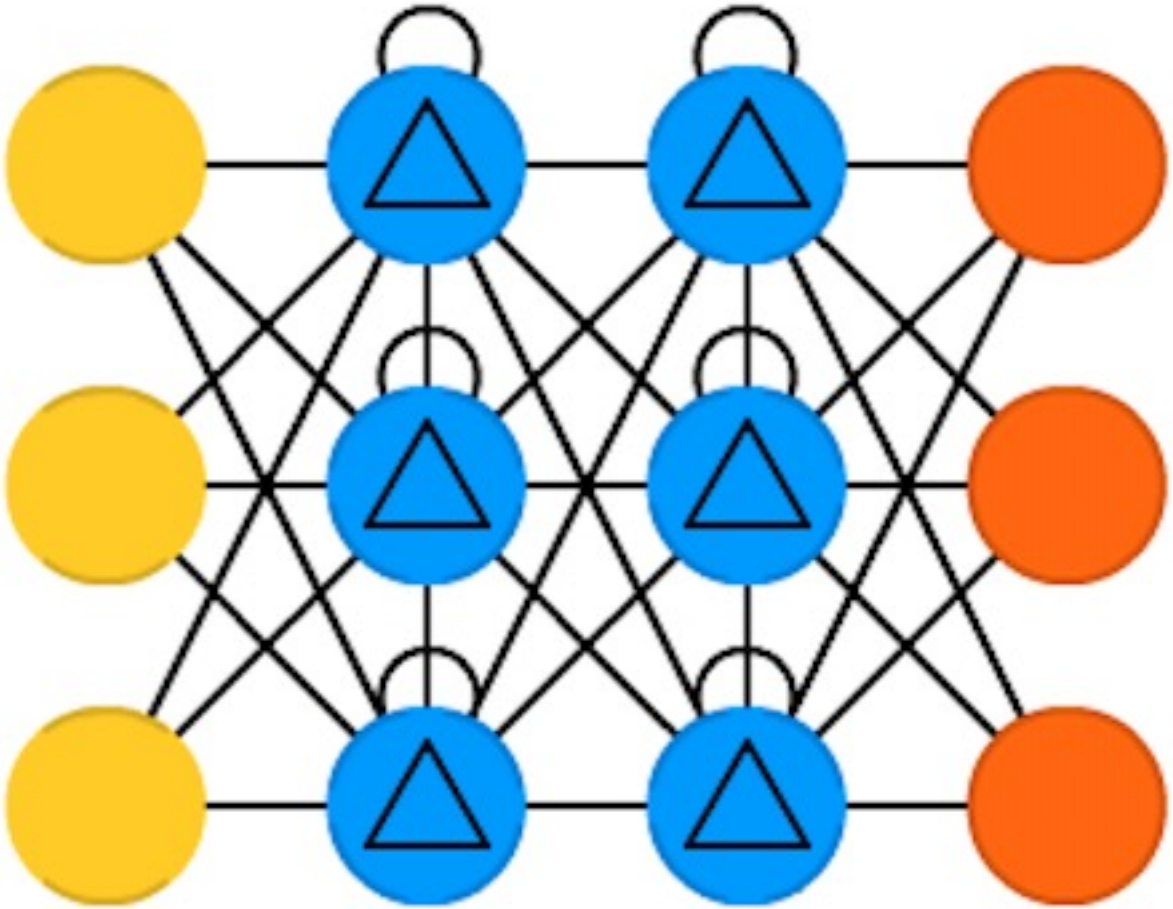
Long / Short Term Memory (LSTM)



Hochreiter, Sepp, and Jürgen Schmidhuber. “Long short-term memory.” Neural computation 9.8 (1997): 1735-1780.

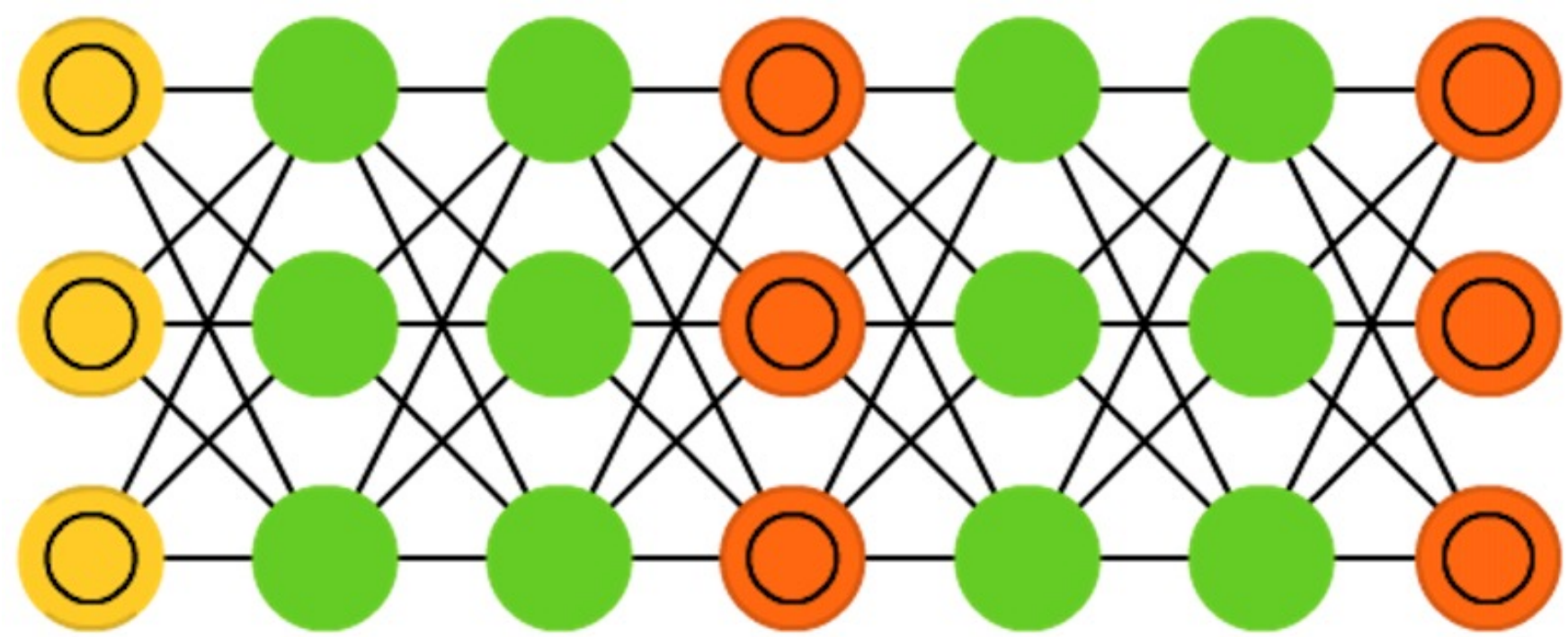
Source: http://www.asimovinstitute.org/neural-network-zoo/

Gated Recurrent Units (GRU)



Chung, Junyoung, et al. “Empirical evaluation of gated recurrent neural networks on sequence modeling.” arXiv preprint arXiv:1412.3555 (2014).
Source: http://www.asimovinstitute.org/neural-network-zoo/

Generative Adversarial Networks (GAN)



Goodfellow, Ian, et al. “Generative adversarial nets.” Advances in Neural Information Processing Systems. 2014.

Source: http://www.asimovinstitute.org/neural-network-zoo/

TensorFlow

Tensor

- 3
  - # a rank 0 tensor; this is a scalar with shape []
- [1. ,2., 3.]
  - # a rank 1 tensor; this is a vector with shape [3]
- [[1., 2., 3.], [4., 5., 6.]]
  - # a rank 2 tensor; a matrix with shape [2, 3]
- [[[1., 2., 3.]], [[7., 8., 9.]]]
  - # a rank 3 tensor with shape [2, 1, 3]

https://www.tensorflow.org/

Scalar 80

Vector [50 60 70]

Matrix

$$\begin{bmatrix} 50 & 60 & 70 \\ 55 & 65 & 75 \end{bmatrix}$$

Tensor

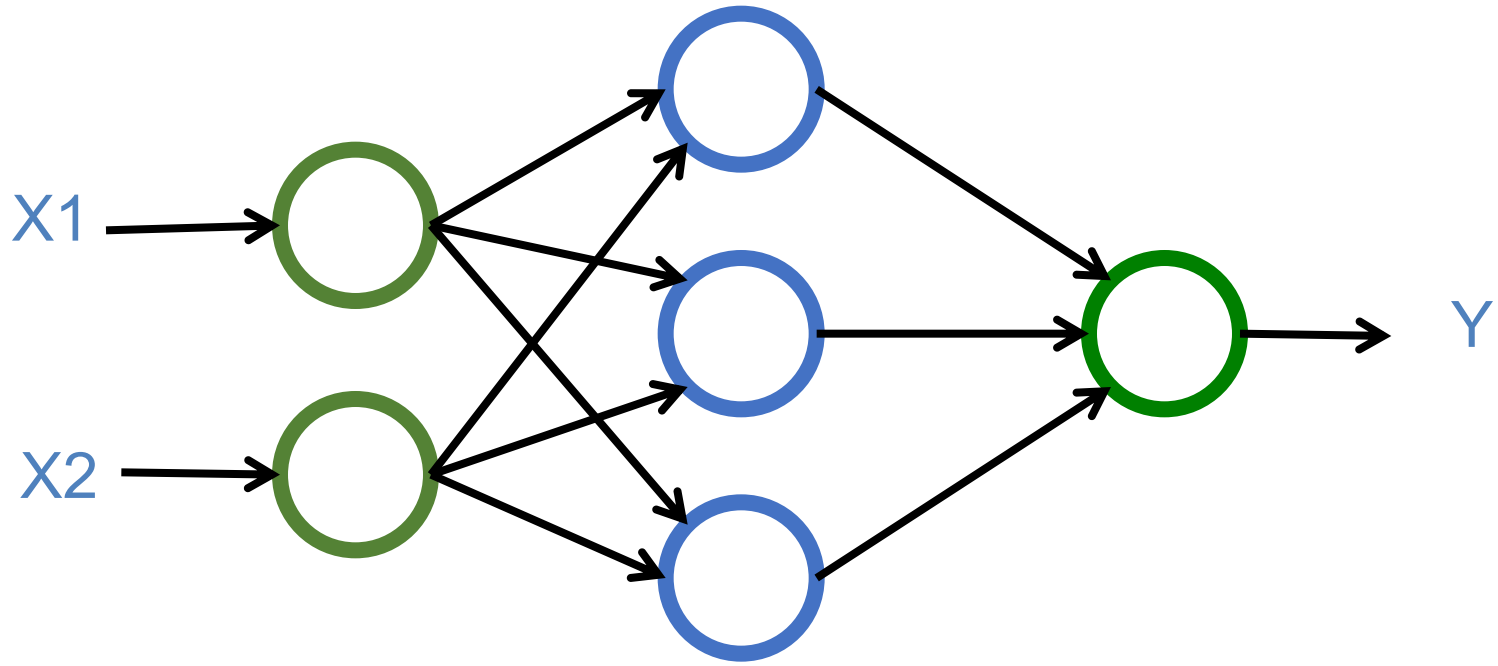
$$\begin{bmatrix} [50 & 60 & 70] & [70 & 80 & 90] \\ [55 & 65 & 75] & [75 & 85 & 95] \end{bmatrix}$$

Deep Learning Foundations:

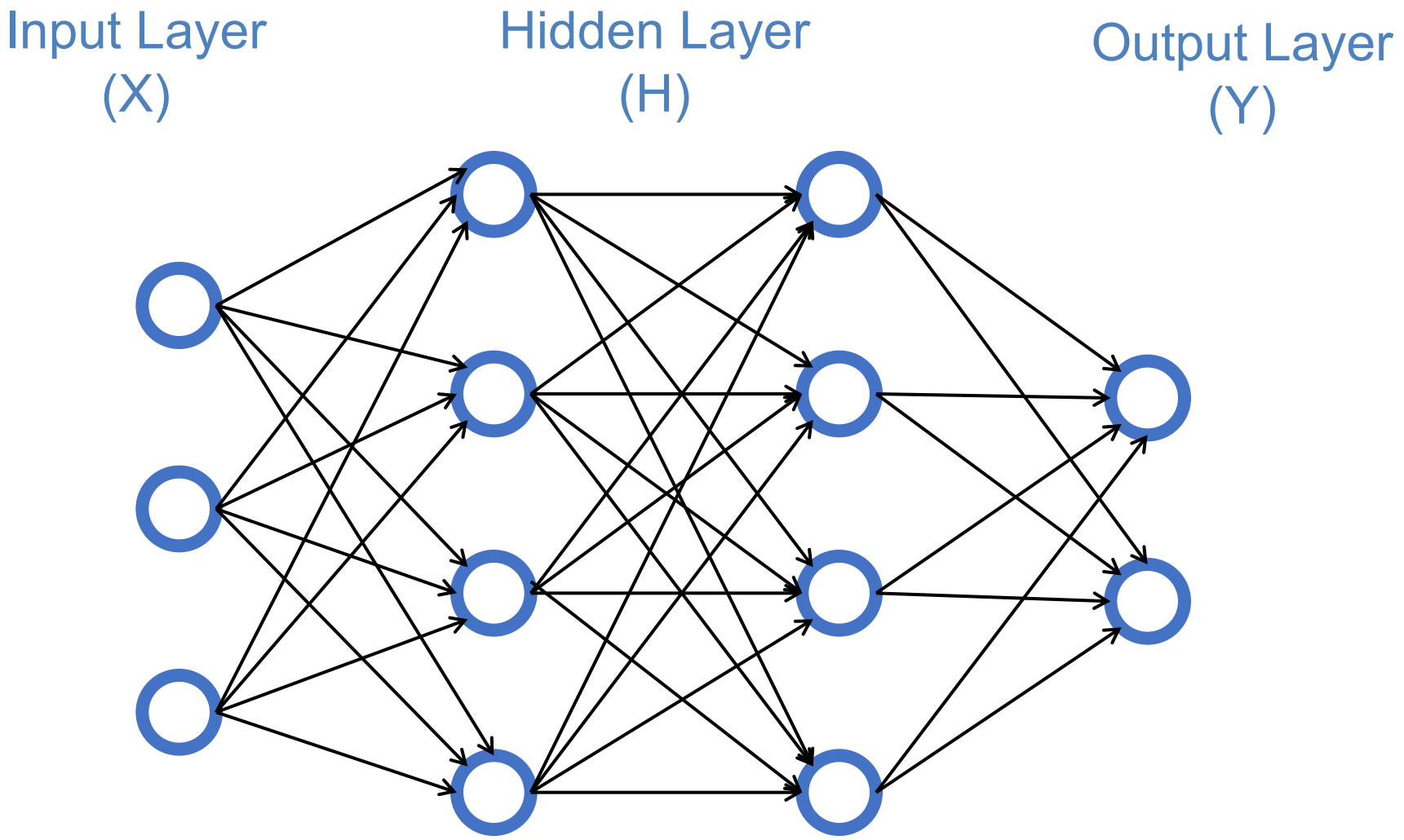
Neural Networks

Deep Learning and Neural Networks

Input Layer (X) Hidden Layer (H) **Output Layer (Y)**



Deep Learning and Neural Networks



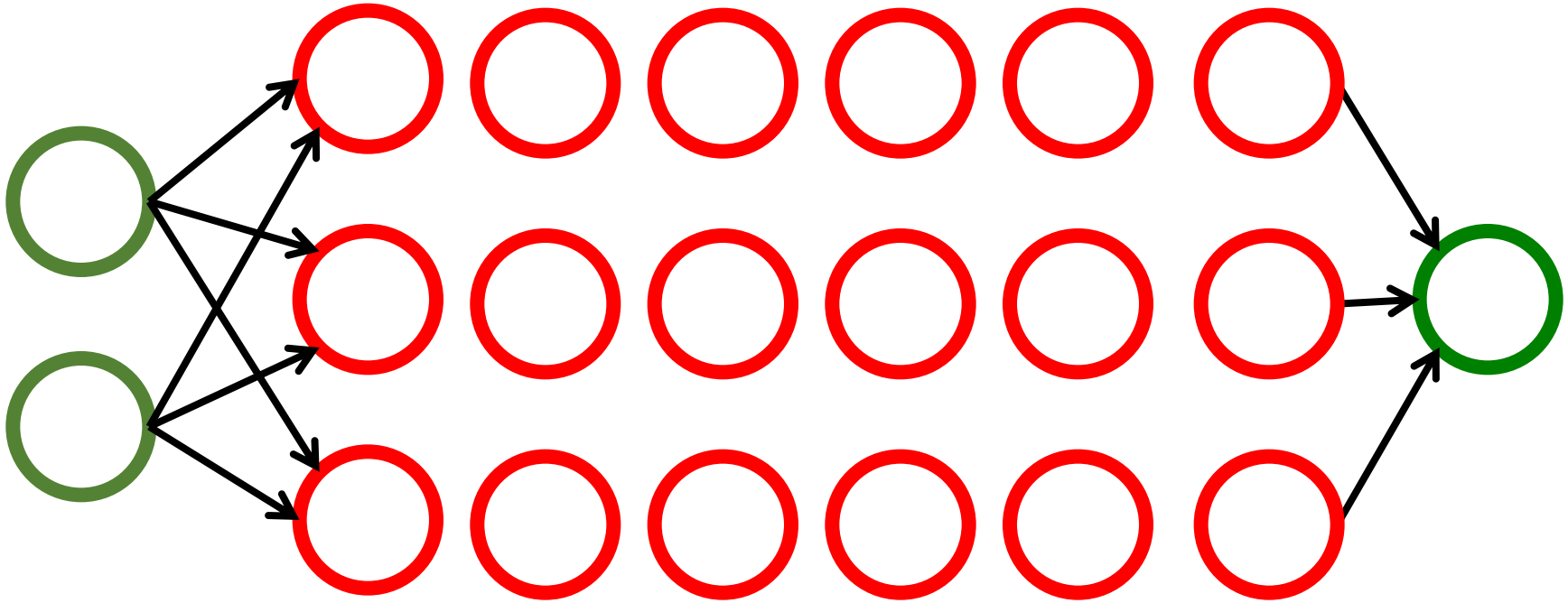
Deep Learning and Neural Networks

Input Layer (X)

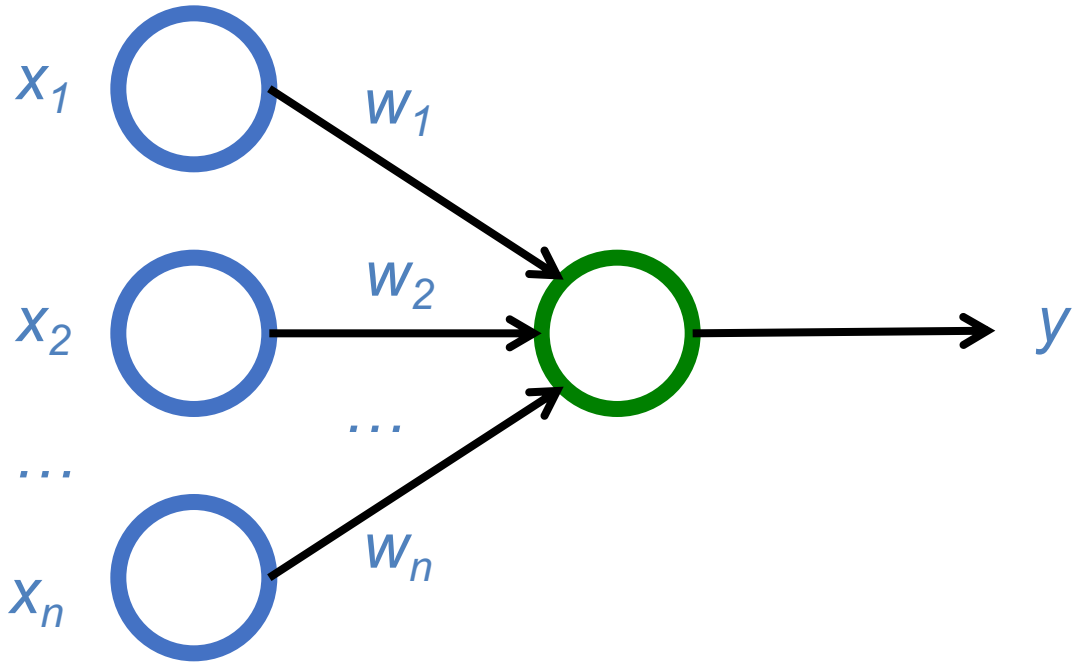
Hidden Layers (H)

Output Layer (Y)

Deep Neural Networks
Deep Learning

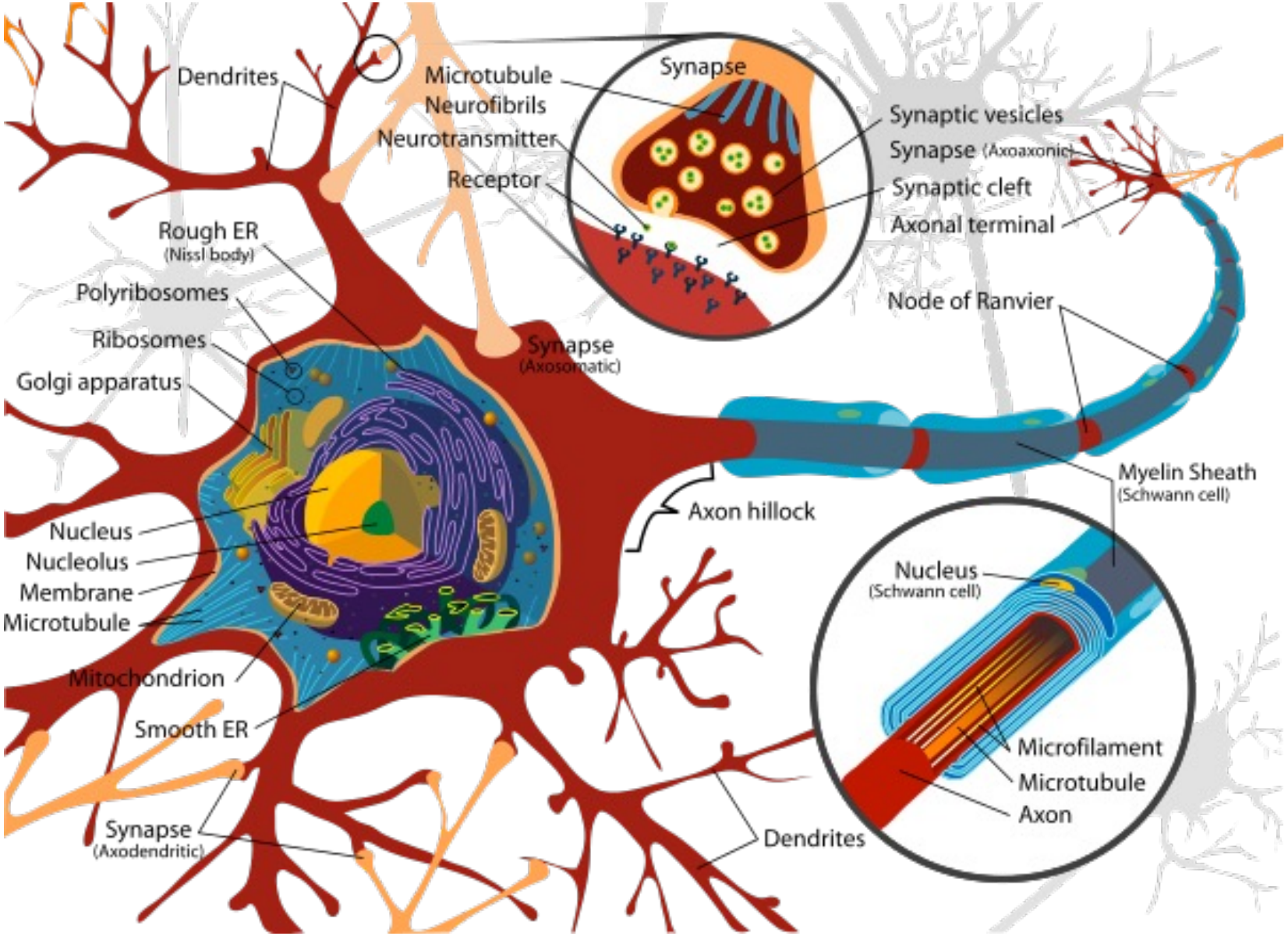


The Neuron



Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

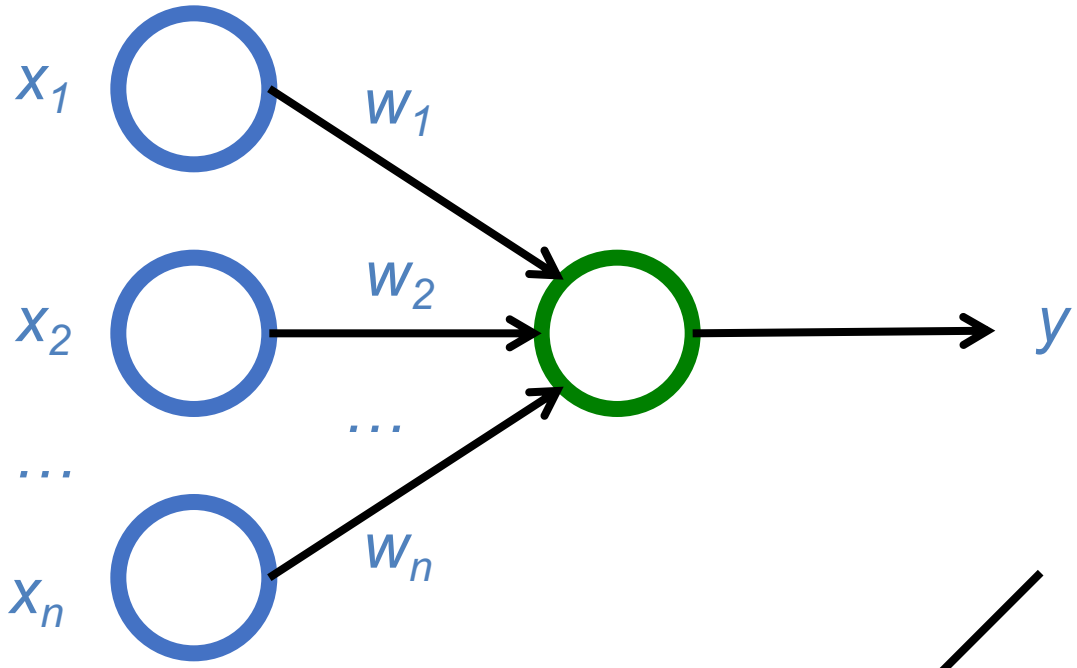
Neuron and Synapse

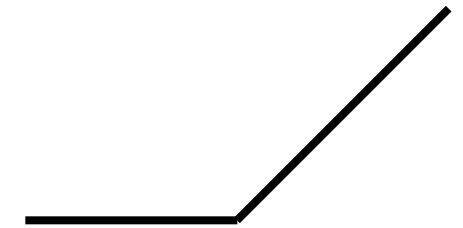


Source: https://en.wikipedia.org/wiki/Neuron

The Neuron

$$y = F\left(\sum_i w_i x_i\right)$$

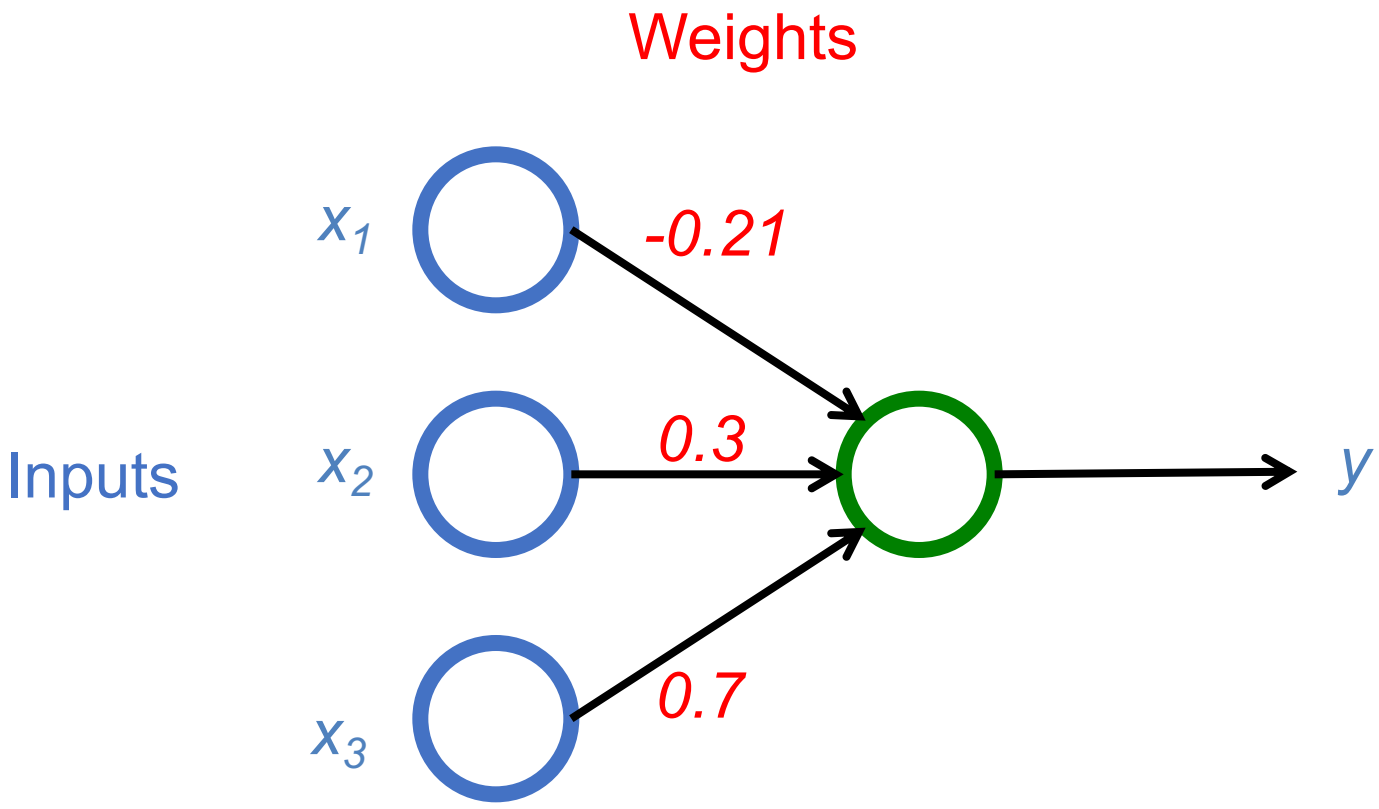




$$F(x) = \max(0, x)$$

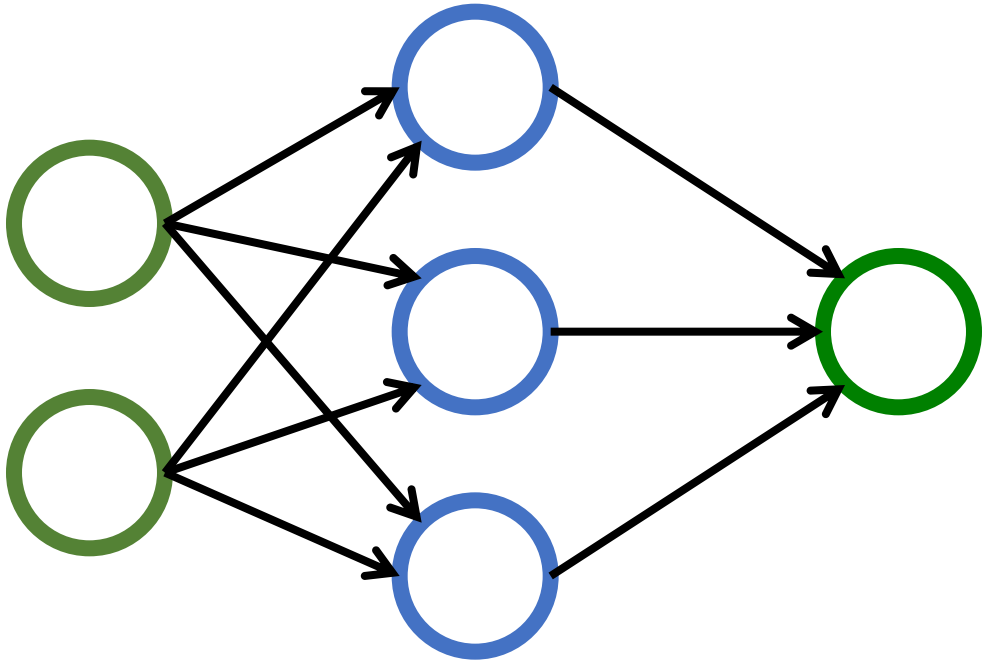
Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

$$y = max\ (\ 0,\ -0.21\ *\ x_1 + 0.3\ *\ x_2 + 0.7\ *\ x_3\ )$$



Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

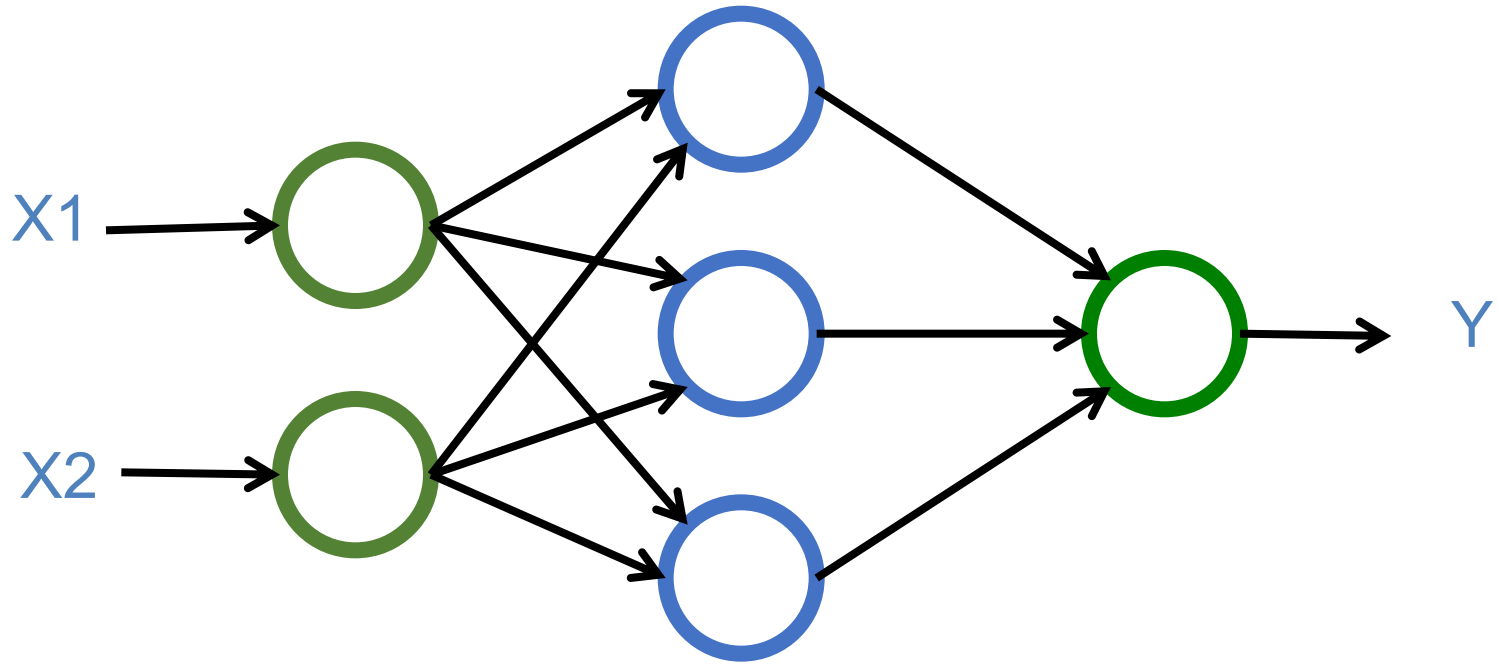
Neural Networks



Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Neural Networks

Input Layer (X) Hidden Layer (H) **Output Layer (Y)**



Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

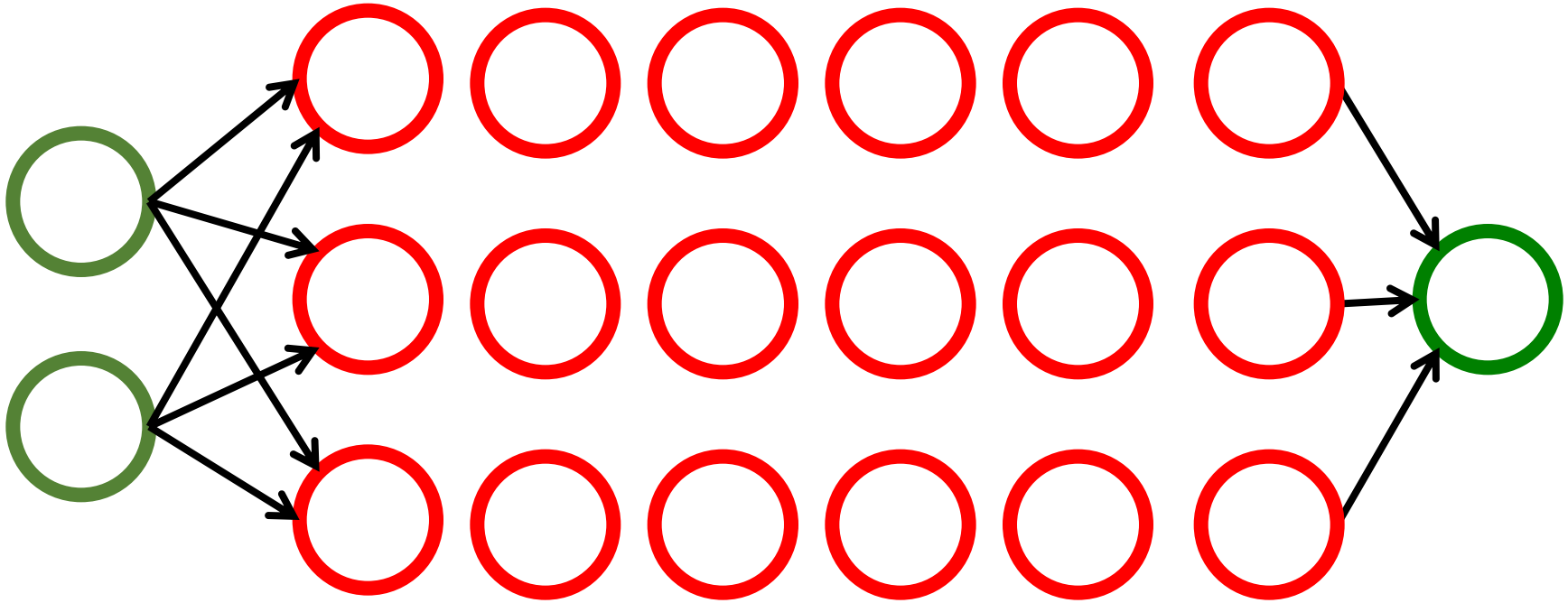
Neural Networks

Input Layer
(X)

Hidden Layers
(H)

Output Layer
(Y)

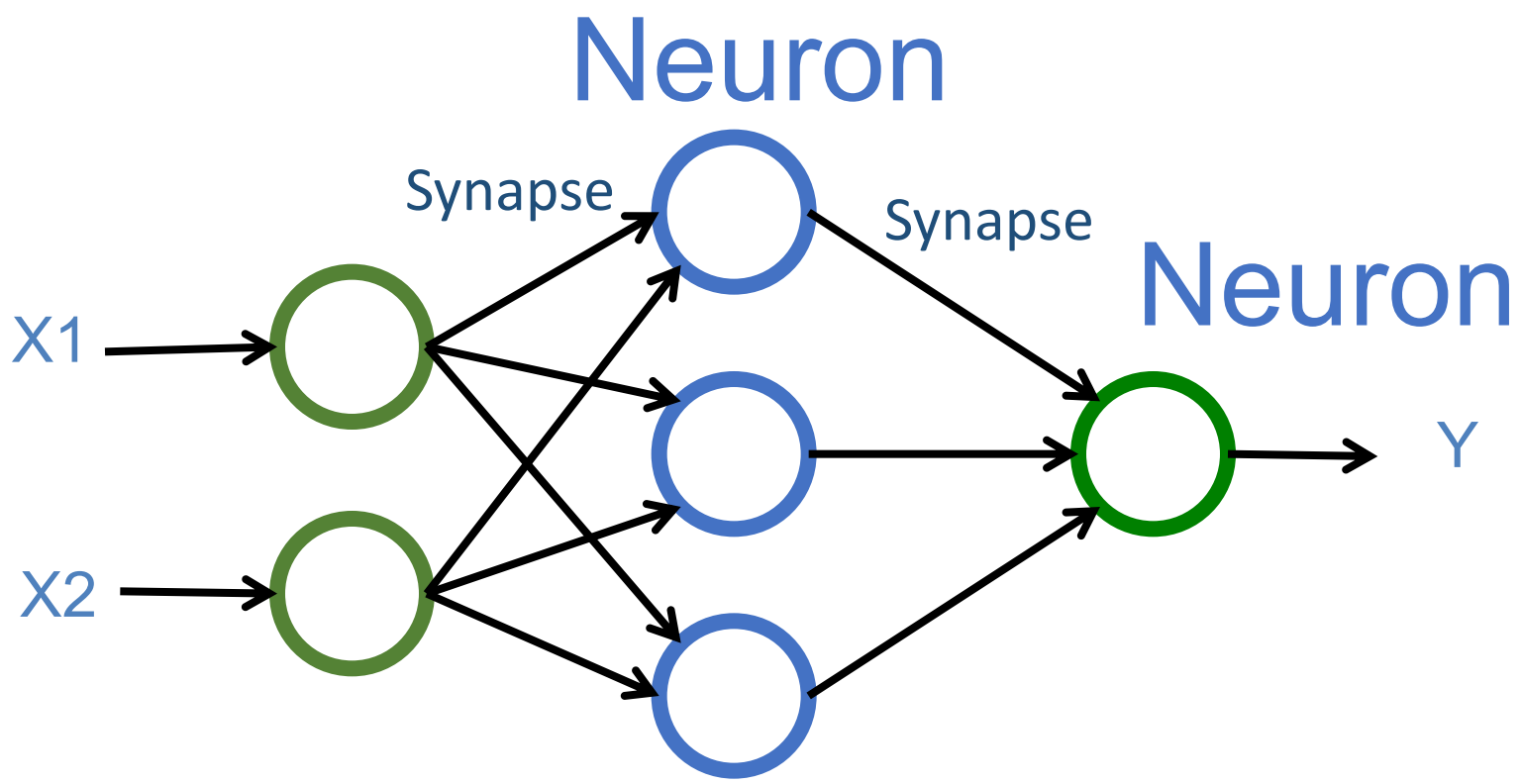
Deep Neural Networks
Deep Learning



Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Neural Networks

Input Layer (X) Hidden Layer (H) **Output Layer (Y)**



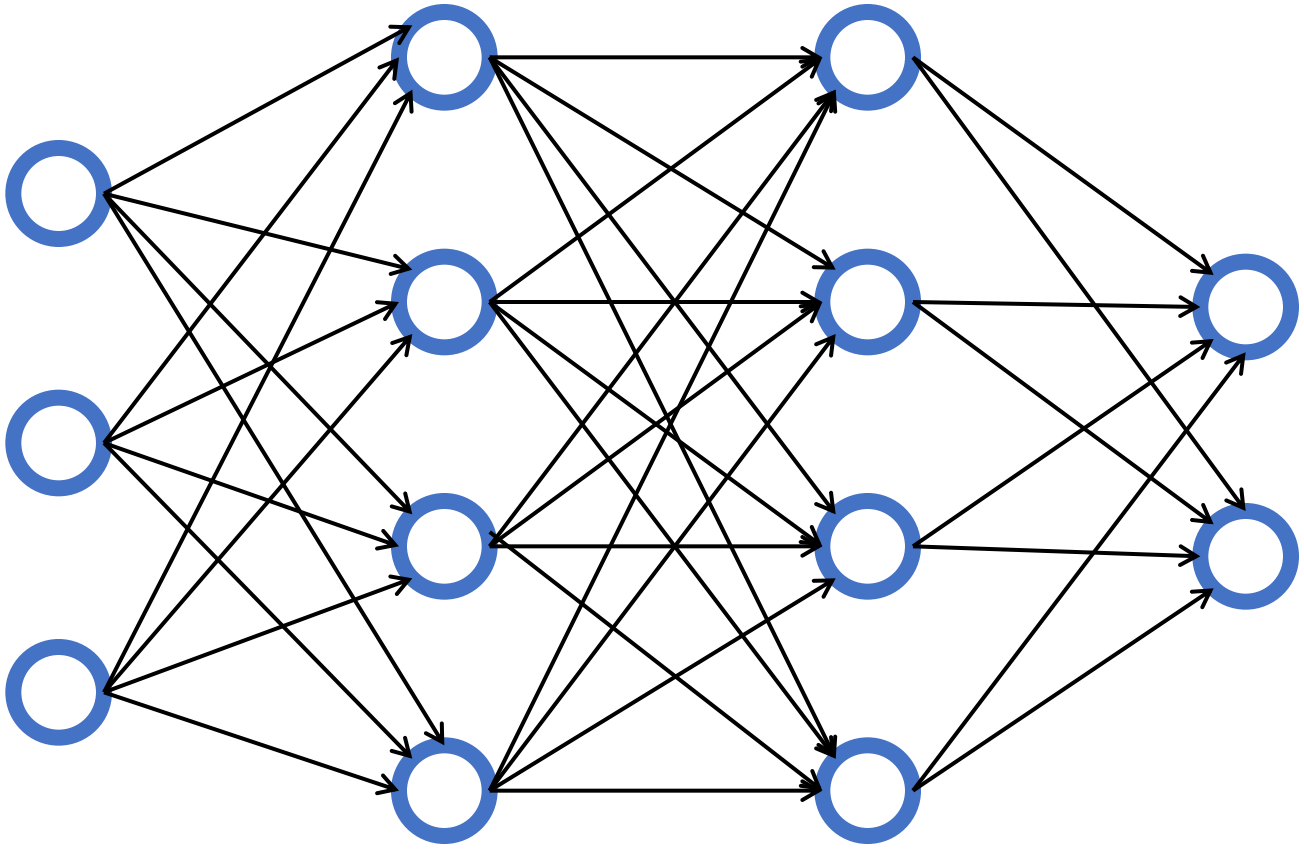
Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Neural Networks

Input Layer
(X)

Hidden Layer
(H)

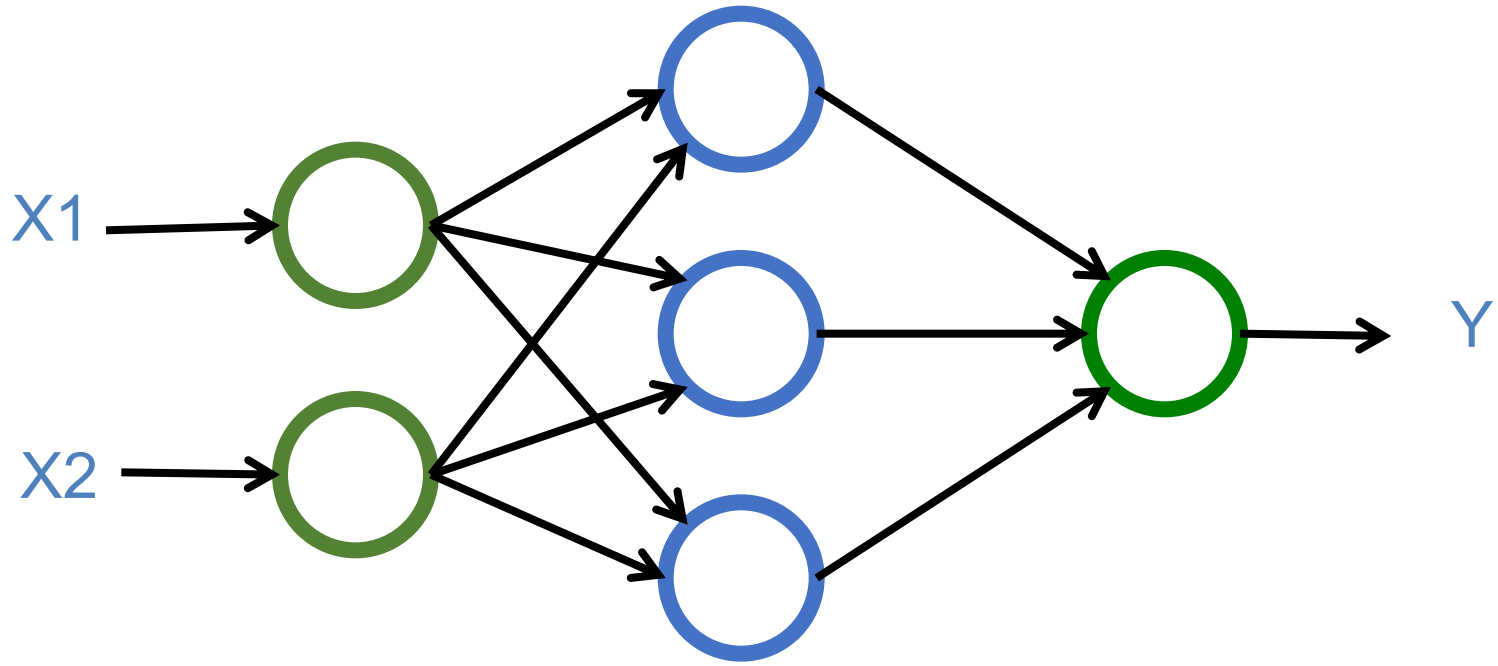
Output Layer
(Y)



Source: https://www.youtube.com/watch?v=P2HPcj8lRJE&list=PLjJh1vlSEYgvGod9wWiydumYl8hOXixNu&index=2

Neural Networks

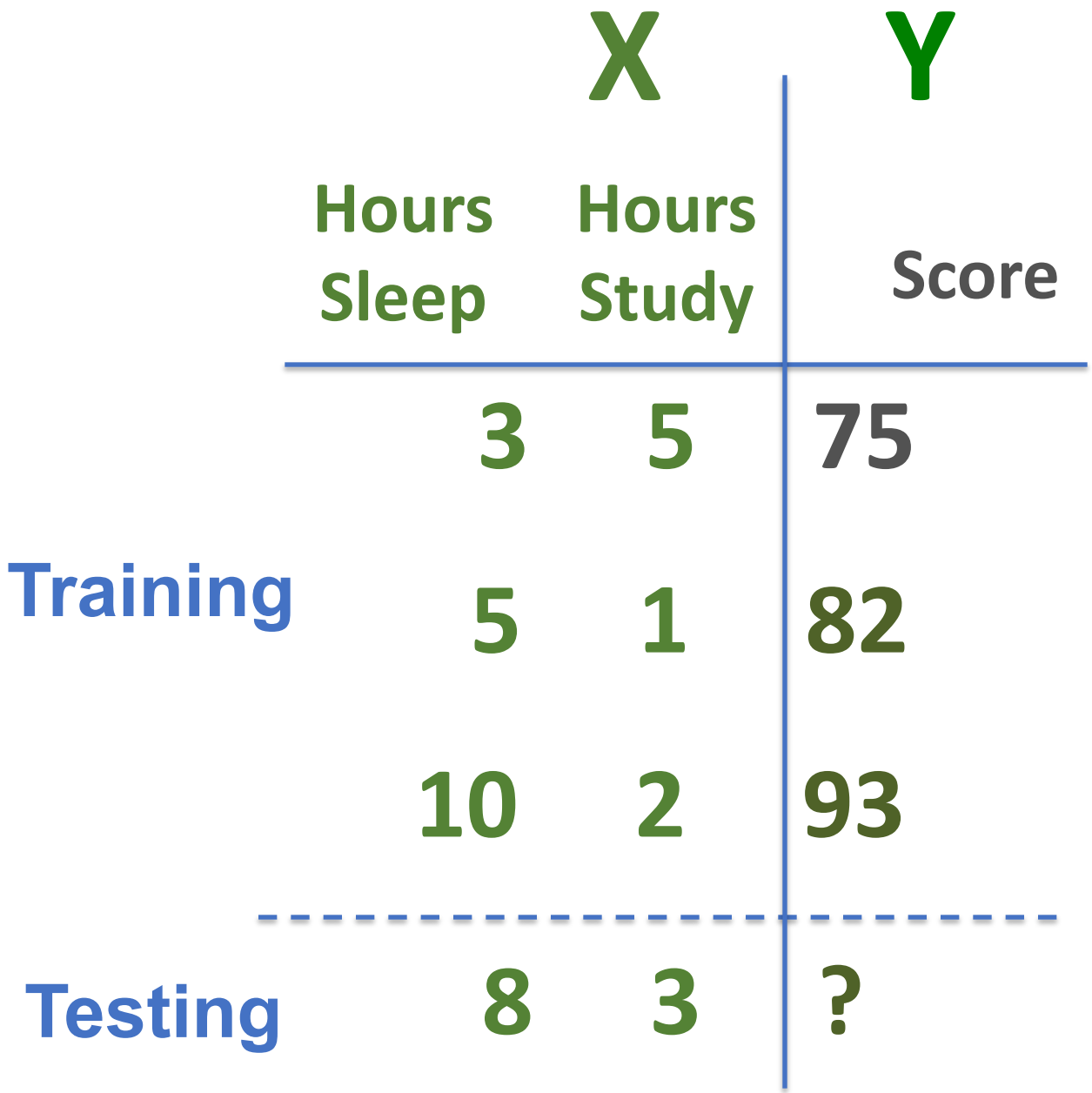
Input Layer (X) Hidden Layer (H) **Output Layer (Y)**



Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

X		Y
Hours Sleep	Hours Study	Score
3	5	75
5	1	82
10	2	93
8	3	?

Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU



Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Linear function

$$y = f(x)$$

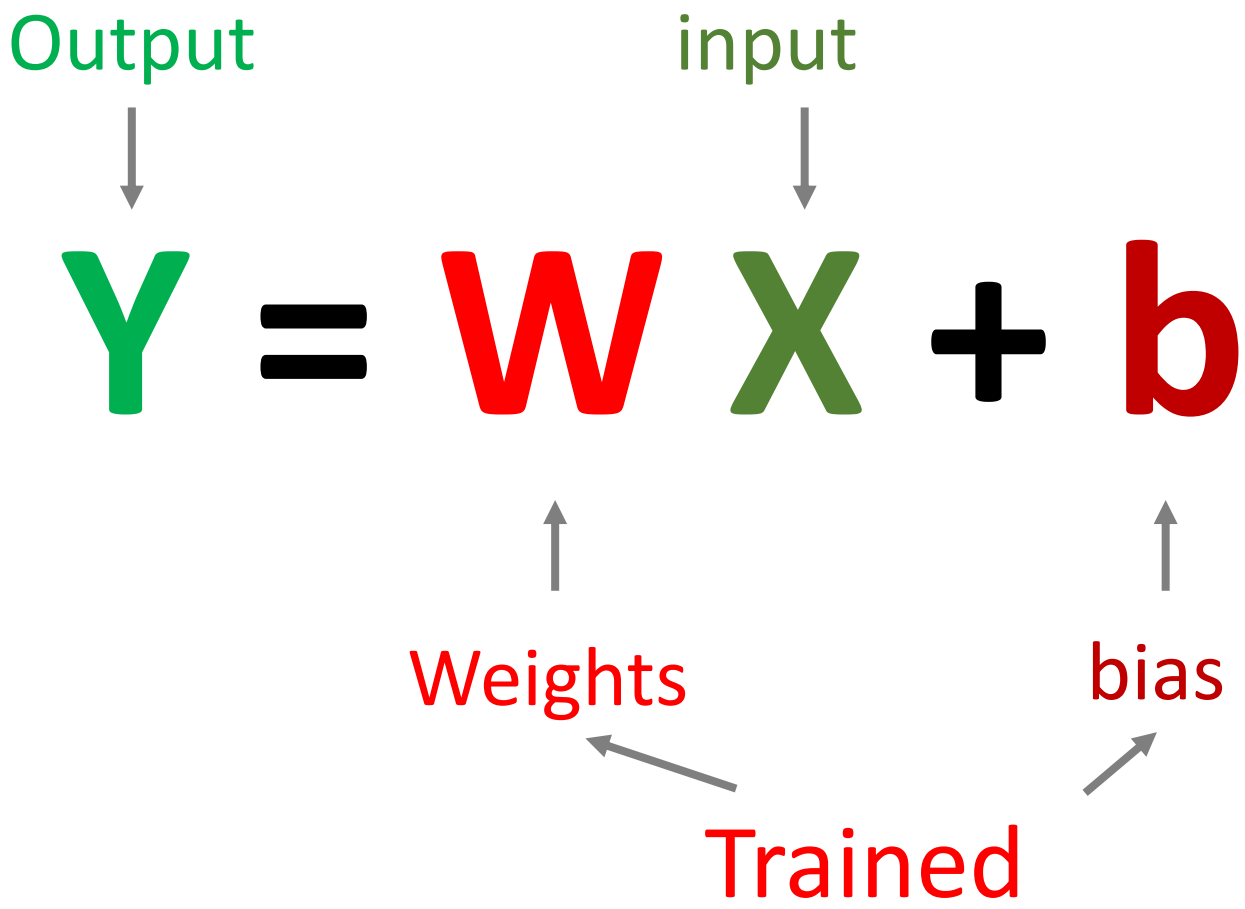
$$y = w_1 x + w_0$$

$$h_w(x) = w_1 x + w_0$$

Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

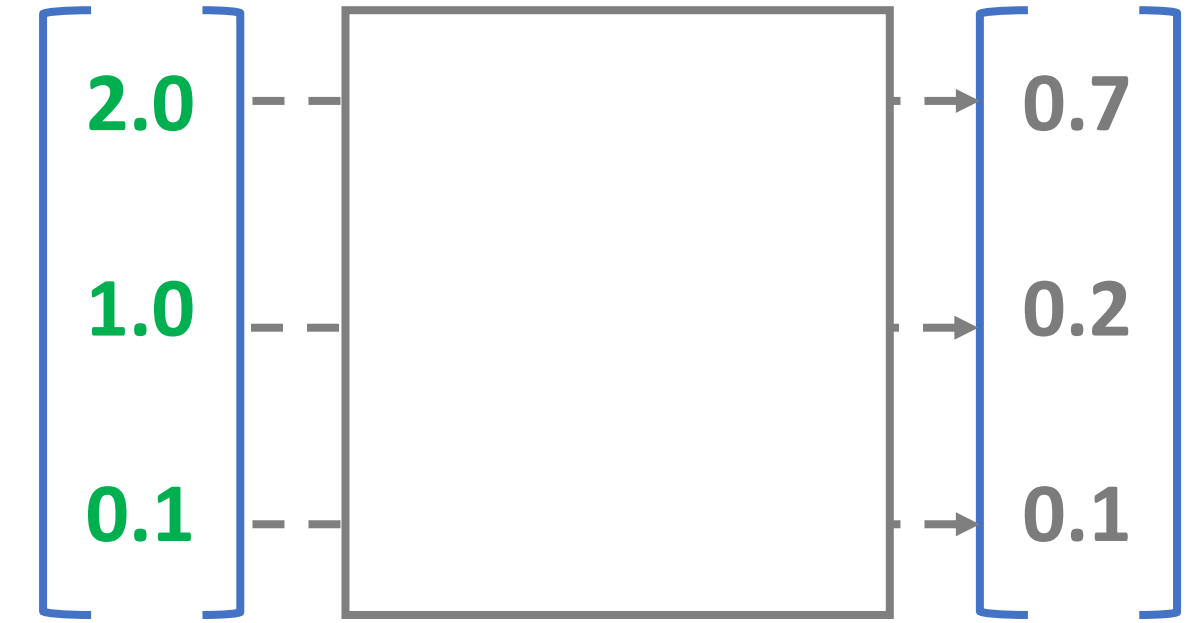
$$Y = W X + b$$

Source: https://www.youtube.com/watch?v=G8eNWzxOgqE



Source: https://www.youtube.com/watch?v=G8eNWzxOgqE

$$W X + b = Y$$

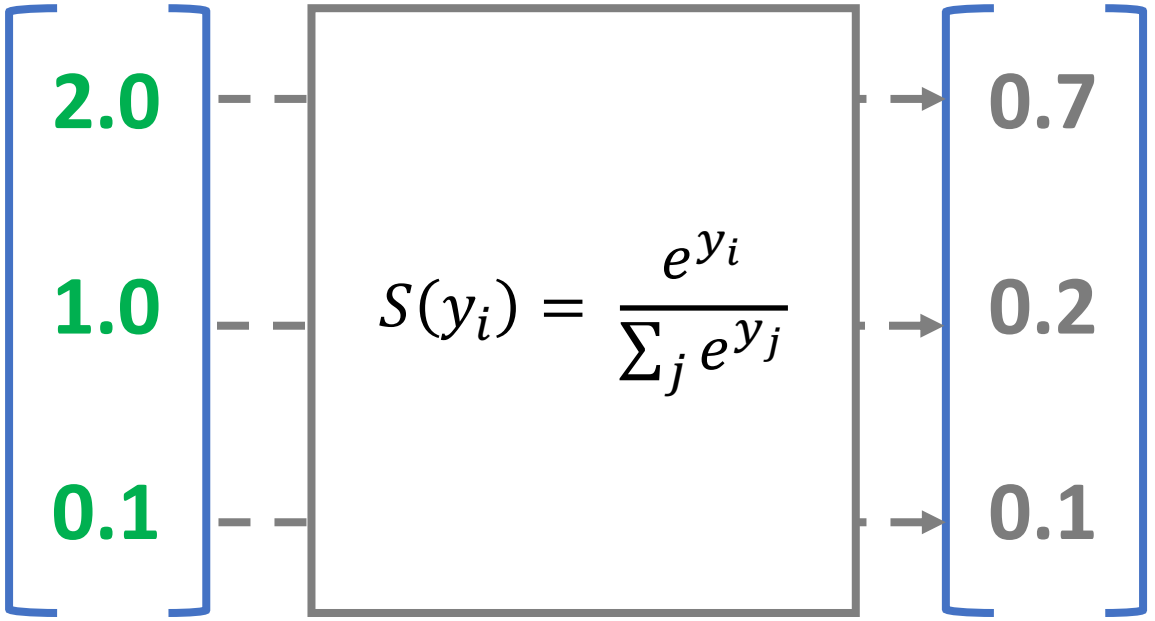


Scores → Probabilities

Source: https://www.youtube.com/watch?v=G8eNWzxOgqE

SoftMAX

$$W\ X + b = Y$$



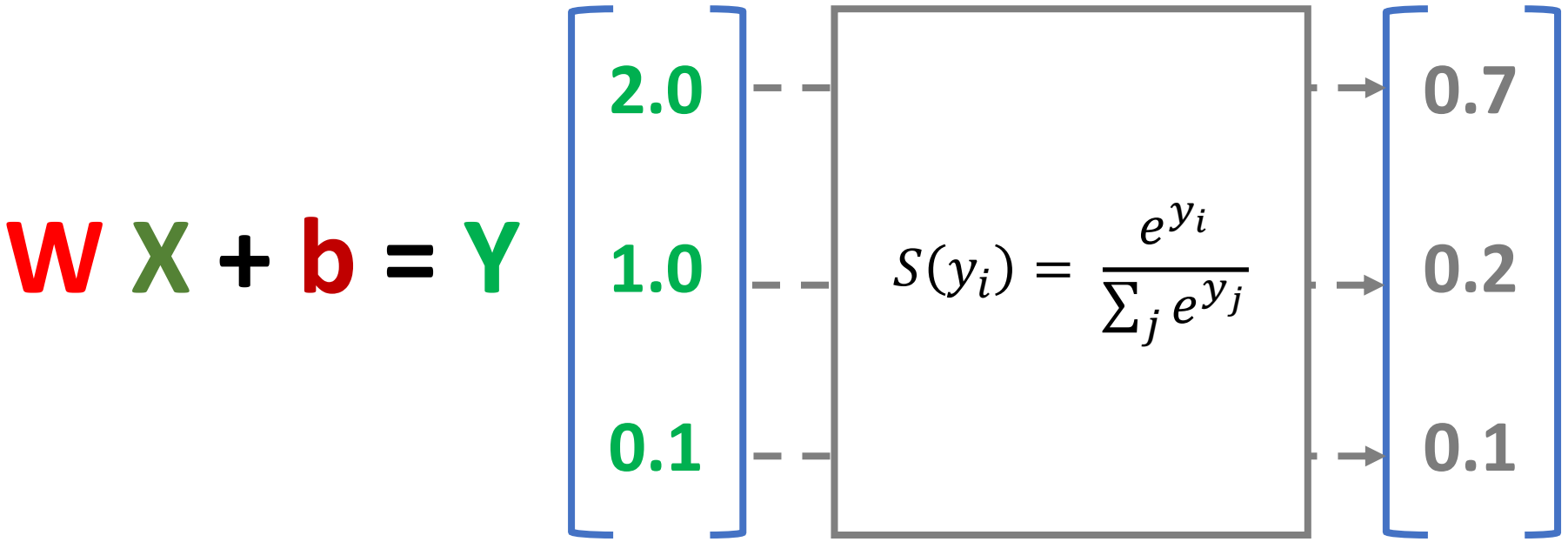
Logits Scores → Probabilities

Source: https://www.youtube.com/watch?v=G8eNWzxOgqE

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_i}} = \frac{e^{2.0}}{e^{2.0}+e^{1.0}+e^{0.1}} = \frac{2.7182^{2.0}}{2.7182^{2.0}+2.7182^{1.0}+2.7182^{0.1}} = 0.7$$

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} = \frac{e^{1.0}}{e^{2.0}+e^{1.0}+e^{0.1}} = \frac{2.7182^{1.0}}{2.7182^{2.0}+2.7182^{1.0}+2.7182^{0.1}} = 0.2$$

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} = \frac{e^{0.1}}{e^{2.0}+e^{1.0}+e^{0.1}} = \frac{2.7182^{0.1}}{2.7182^{2.0}+2.7182^{1.0}+2.7182^{0.1}} = 0.1$$



Source: https://www.youtube.com/watch?v=G8eNWzxOgqE

Training a Network
=
Minimize the Cost Function

Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Training a Network

=

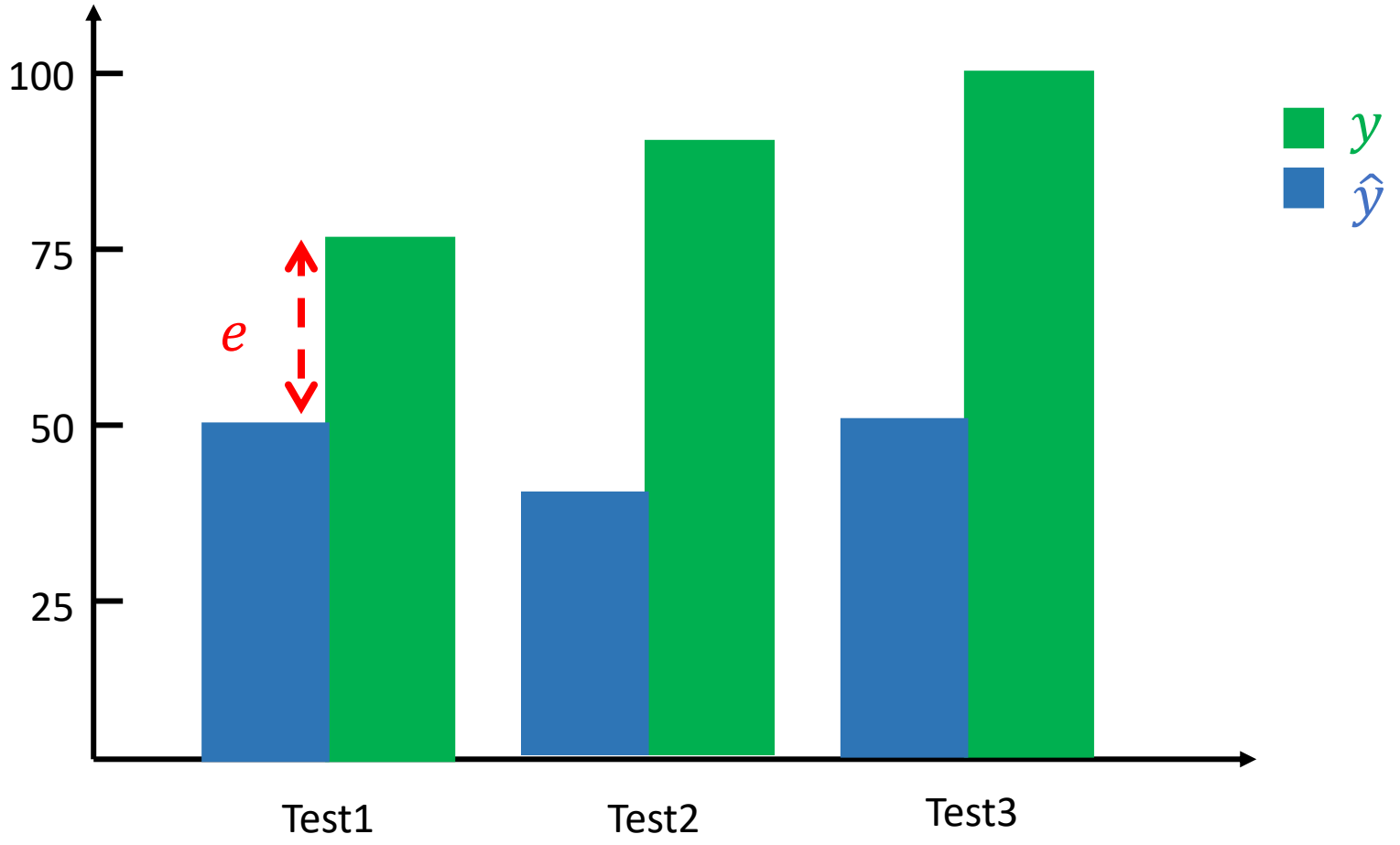
Minimize the Cost Function

Minimize the Loss Function

Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Error = Predict Y - Actual Y

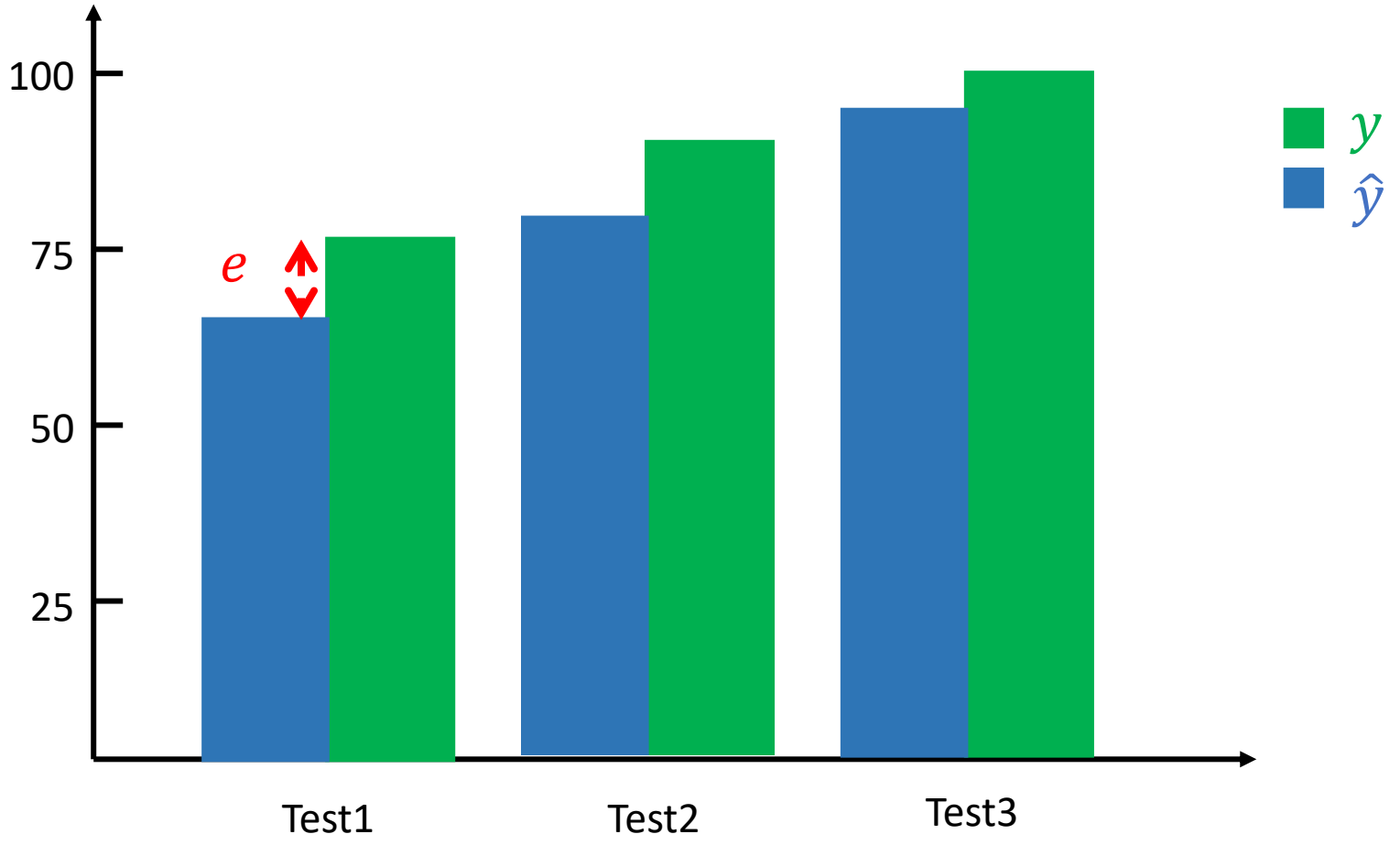
Error : Cost : Loss



Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Error = Predict Y - Actual Y

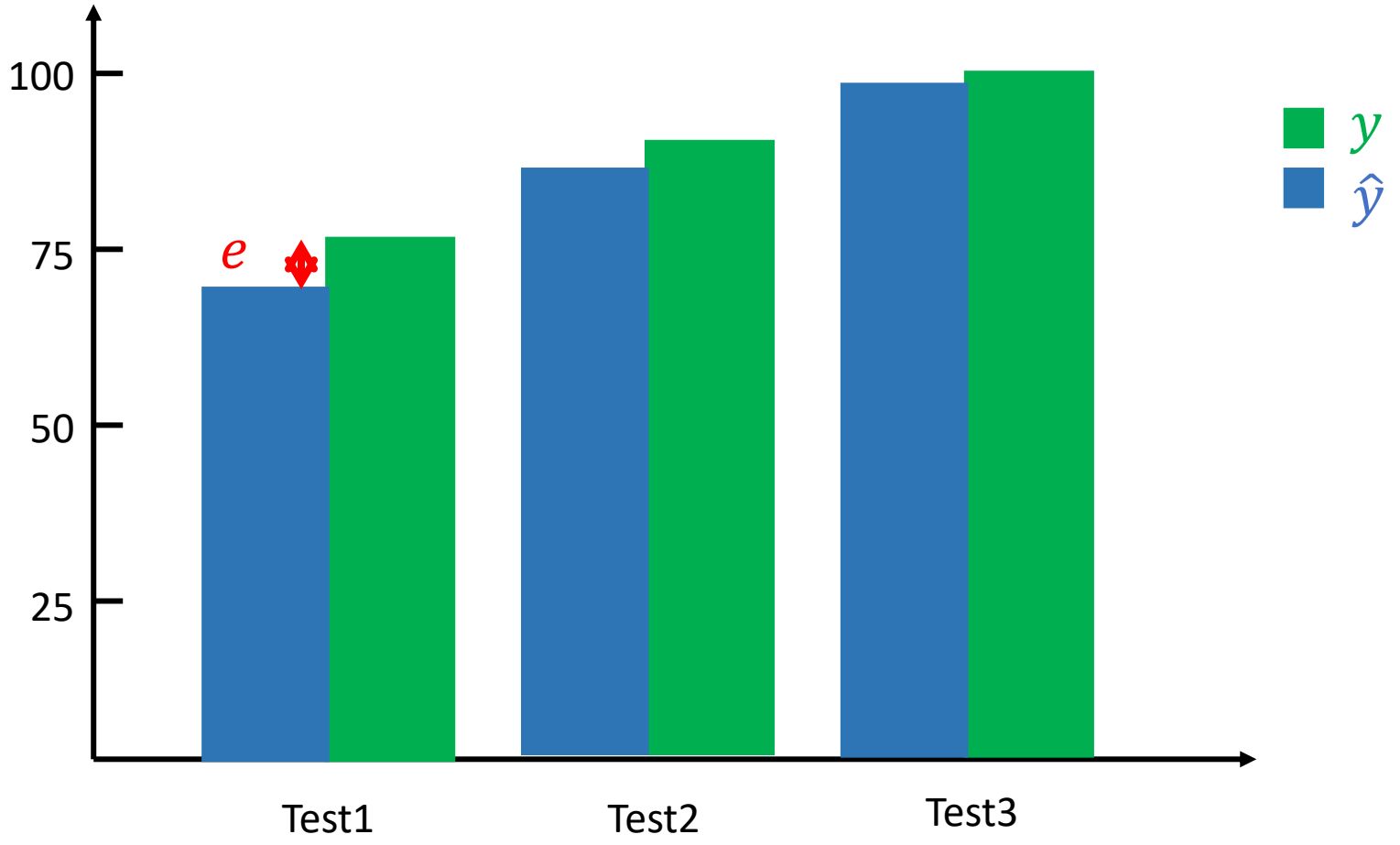
Error : Cost : Loss



Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Error = Predict Y - Actual Y

Error : Cost : Loss

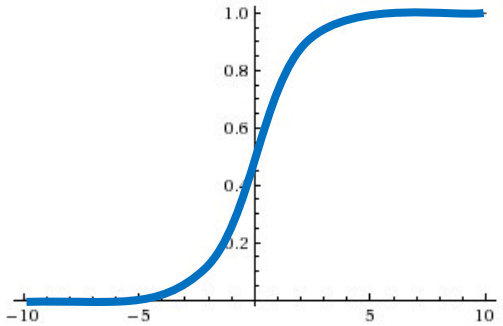


Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Activation Functions

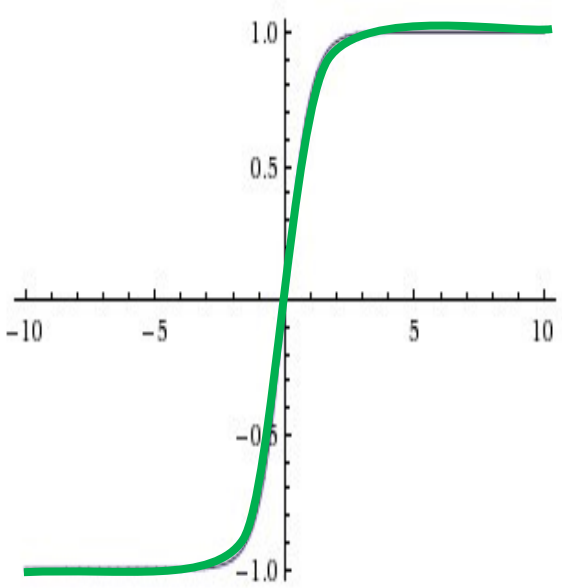
Activation Functions

Sigmoid



[0, 1]

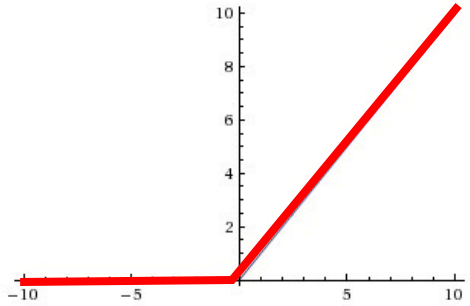
TanH



[-1, 1]

ReLU

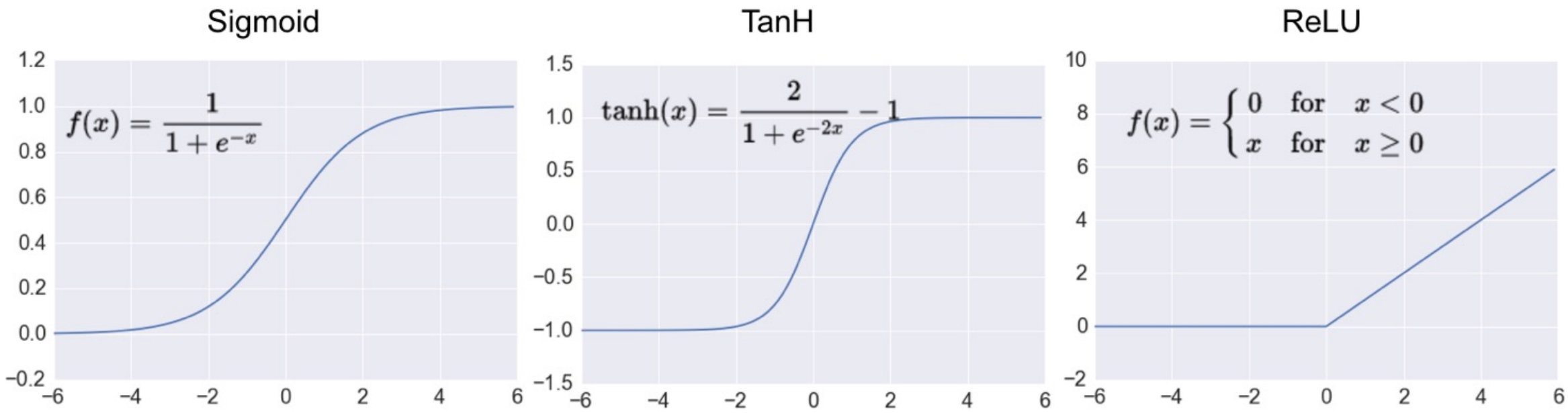
(Rectified Linear Unit)



$f(x) = max(0, x)$

Source: http://cs231n.github.io/neural-networks-1/

Activation Functions



Source: http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/

Loss Function

Binary Classification: 2 Class

Activation Function:
Sigmoid

Loss Function:
Binary Cross-Entropy

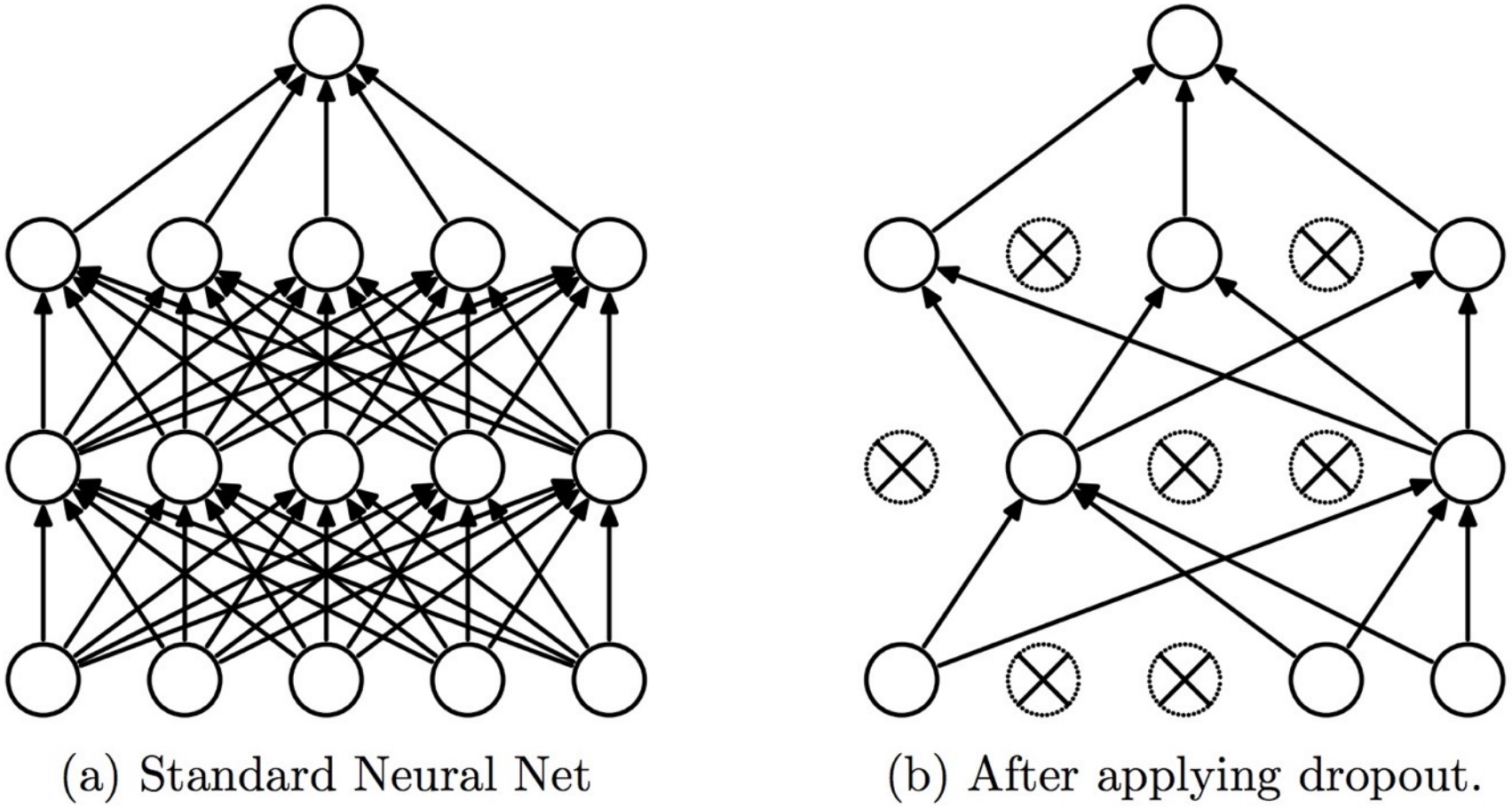
Multiple Classification: 10 Class

Activation Function:
SoftMAX

Loss Function:
Categorical Cross-Entropy

Dropout

Dropout: a simple way to prevent neural networks from overfitting



(a) Standard Neural Net

(b) After applying dropout.

Source: Srivastava, Nitish, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov.
"Dropout: a simple way to prevent neural networks from overfitting." *Journal of machine learning research* 15, no. 1 (2014): 1929-1958.

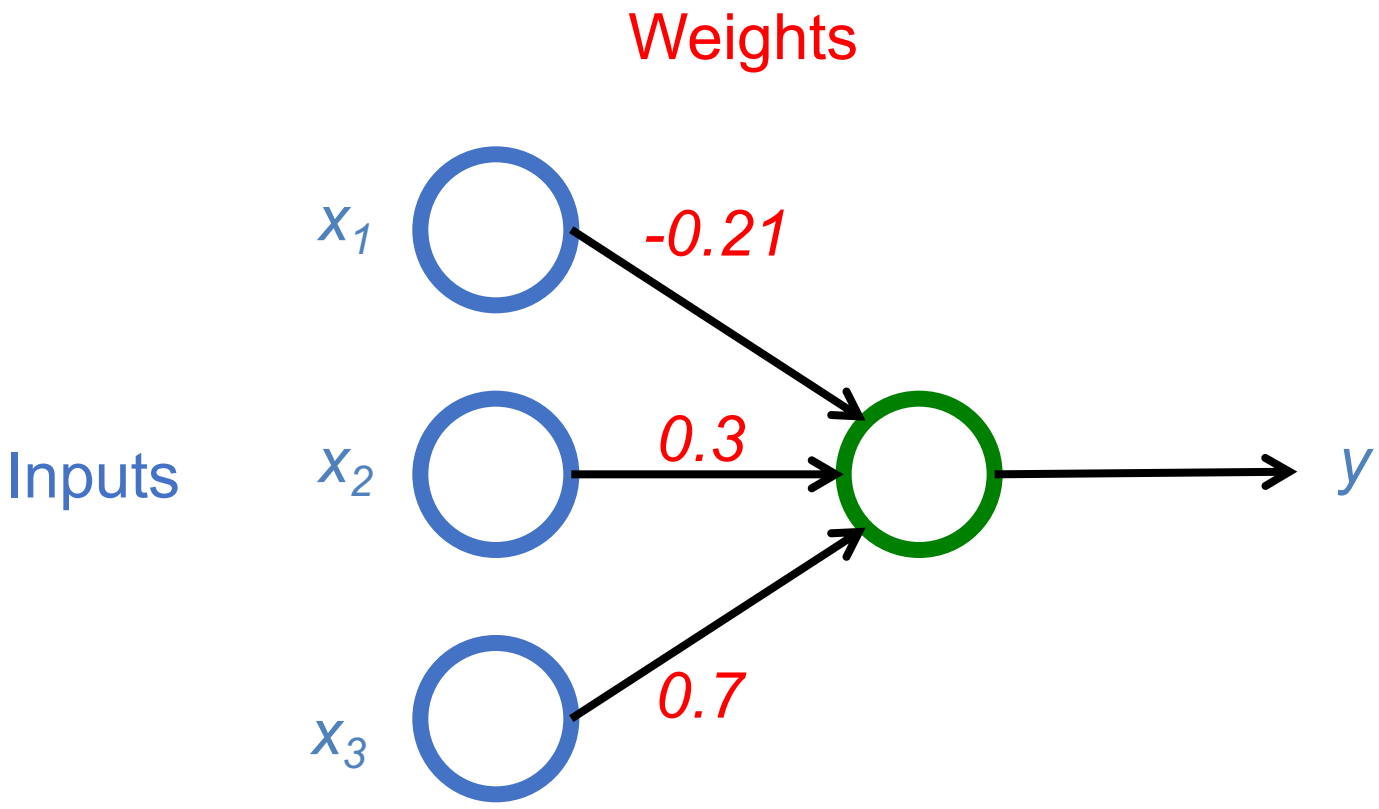
Learning Algorithm

While not done:

- **Pick a random training example "(input, label)"**
- **Run neural network on "input"**
- **Adjust weights on edges to make output closer to "label"**

Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

$$y = max\ (\ 0,\ -0.21\ *\ x_1 + 0.3\ *\ x_2 + 0.7\ *\ x_3\ )$$



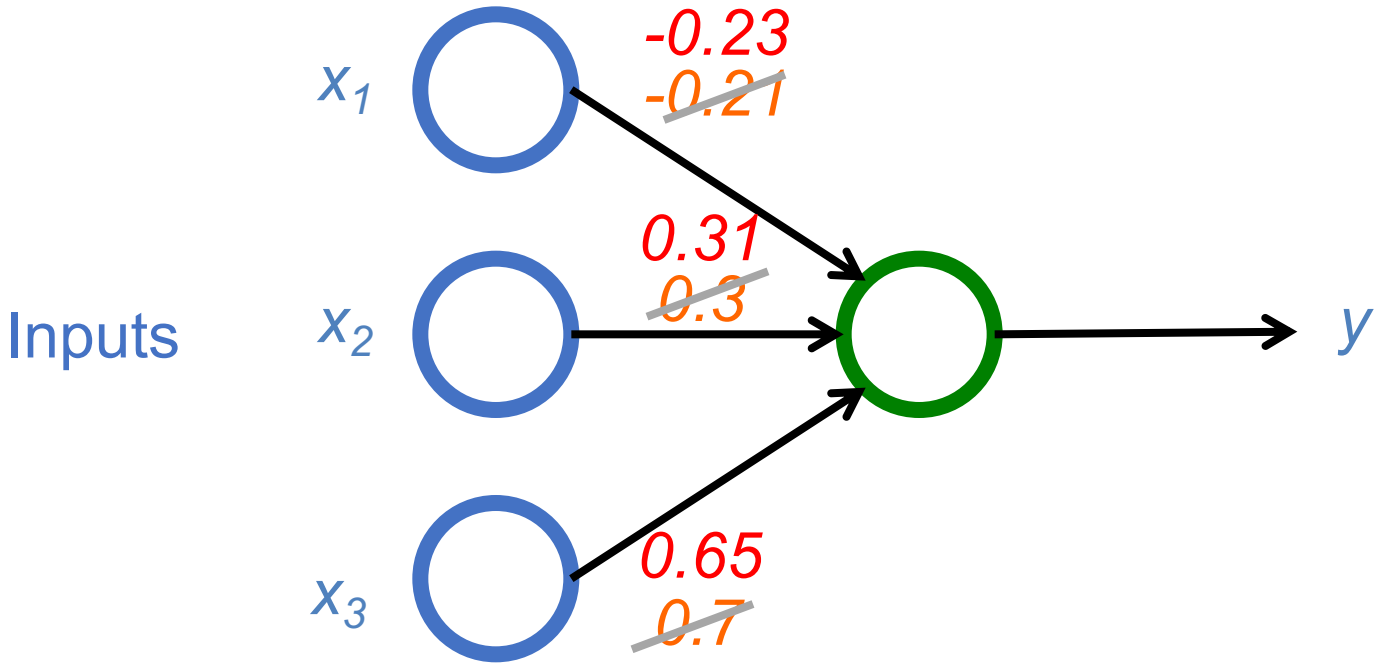
Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

Next time:

$$y = max(0, -0.23 * x_1 + 0.31 * x_2 + 0.65 * x_3)$$

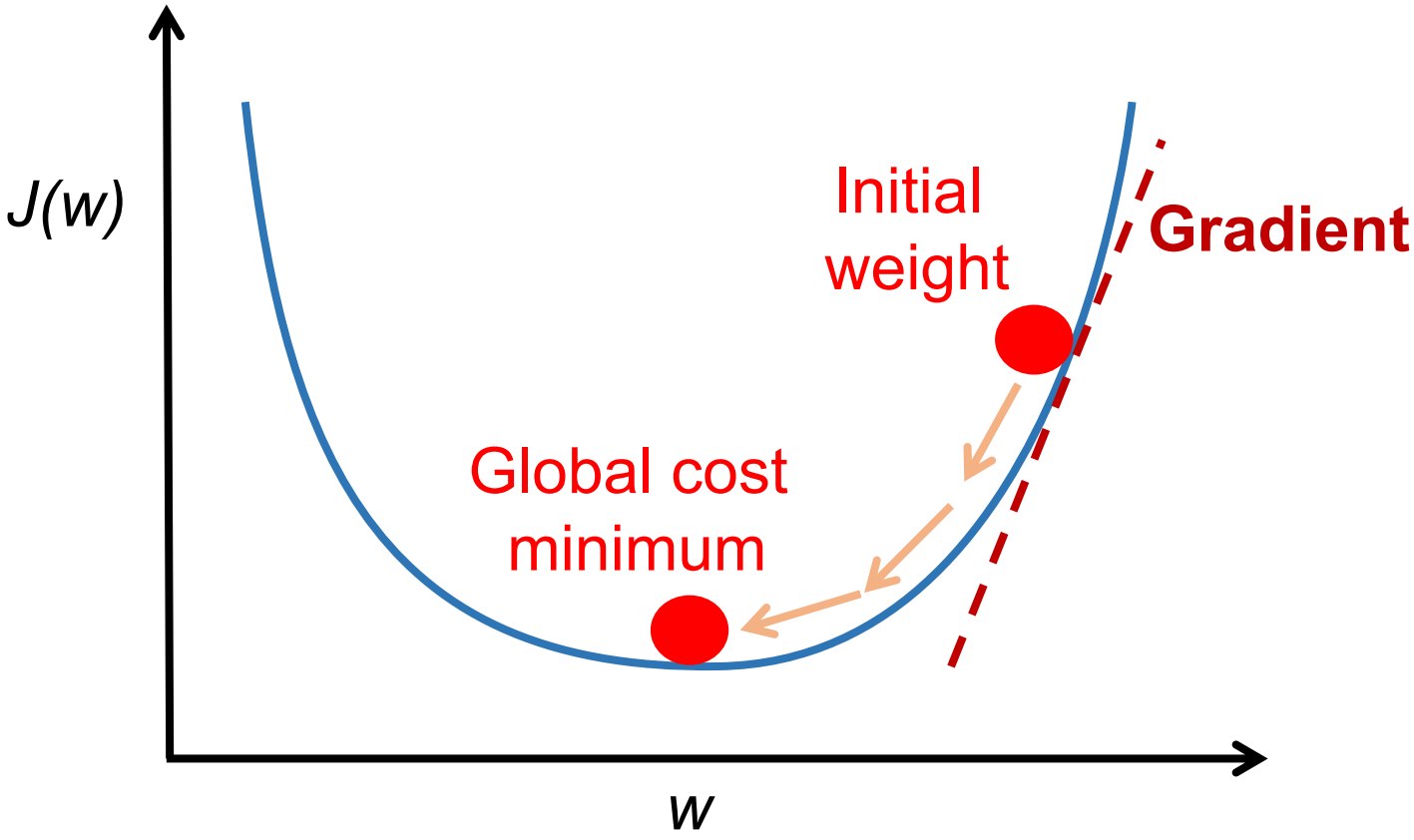
~~$$y = max(0, -0.21 * x_1 + 0.3 * x_2 + 0.7 * x_3)$$~~

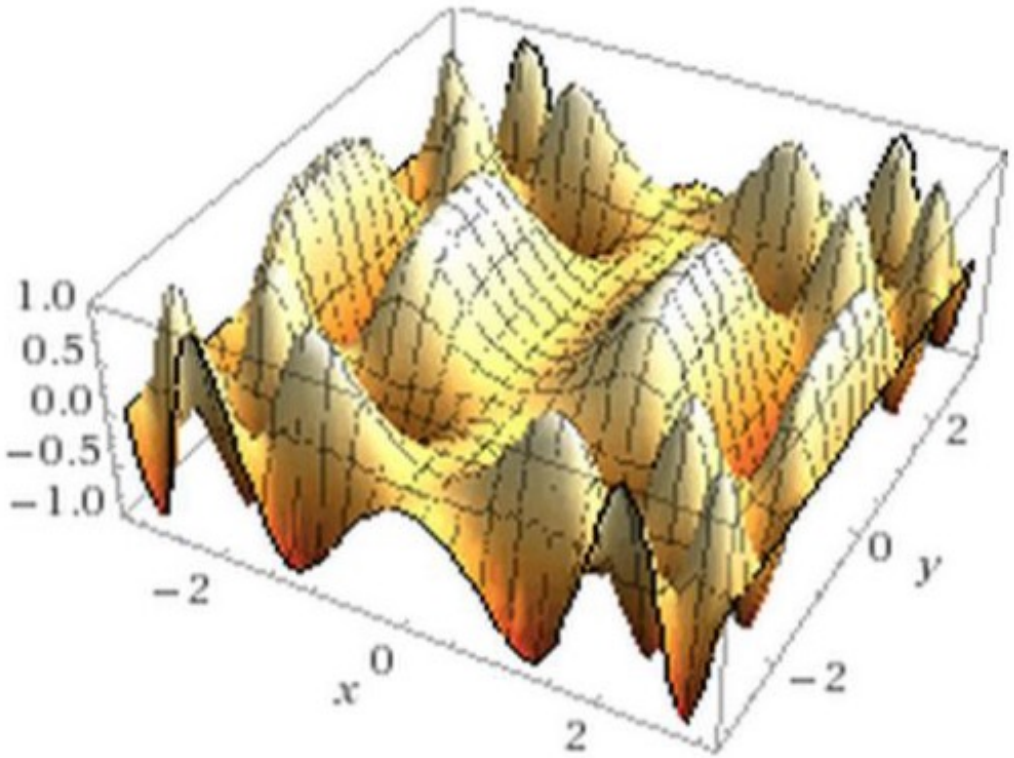
Weights



Optimizer:

Stochastic Gradient Descent (SGD)

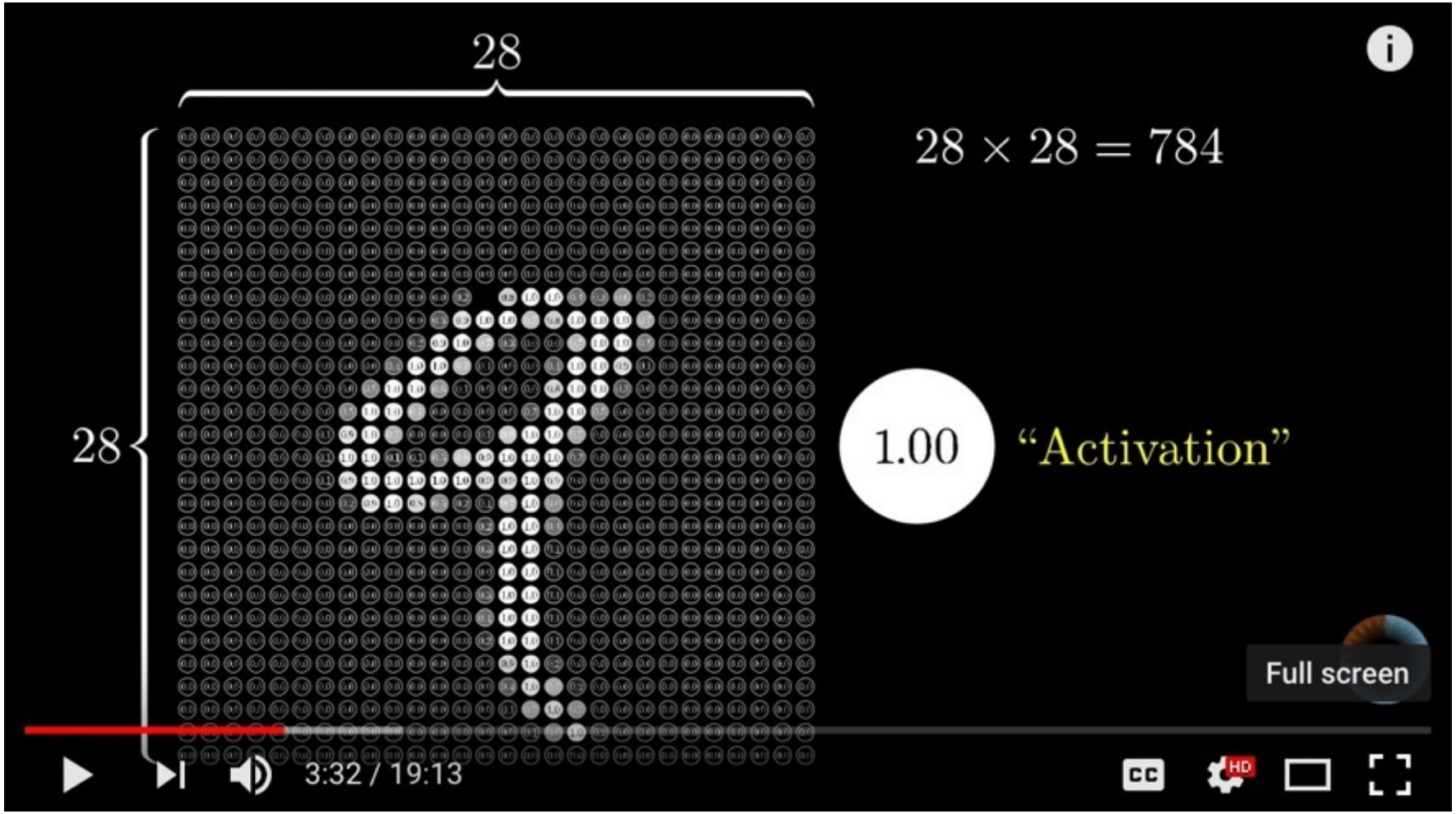




This shows a function of 2 variables: real neural nets are functions of hundreds of millions of variables!

Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

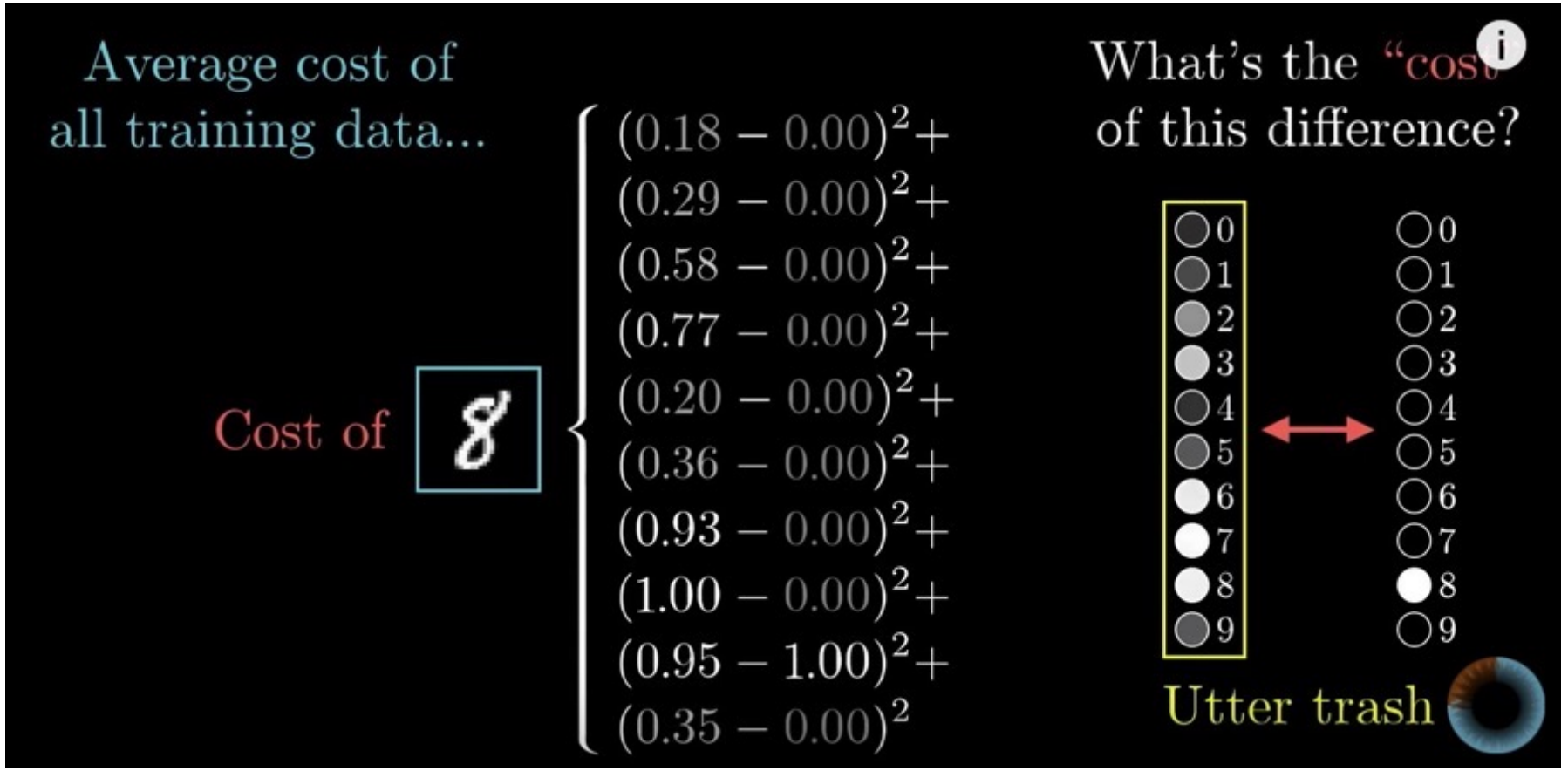
Neural Network and Deep Learning



Source: 3Blue1Brown (2017), But what *is* a Neural Network? | Chapter 1, deep learning, https://www.youtube.com/watch?v=aircAruvnKk

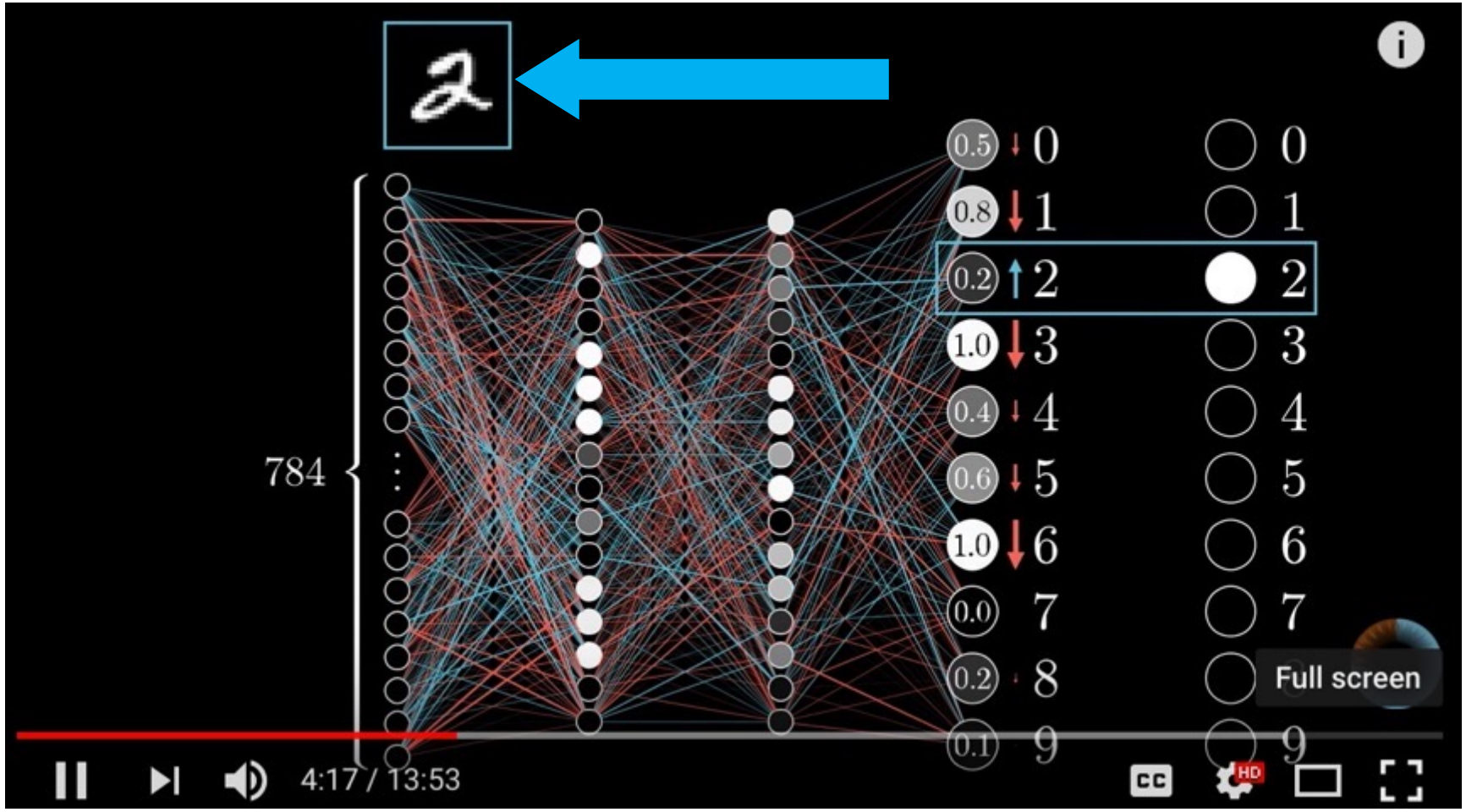
Gradient Descent

how neural networks learn



Source: 3Blue1Brown (2017), Gradient descent, how neural networks learn | Chapter 2, deep learning, https://www.youtube.com/watch?v=IHZwWFHWa-w

Backpropagation



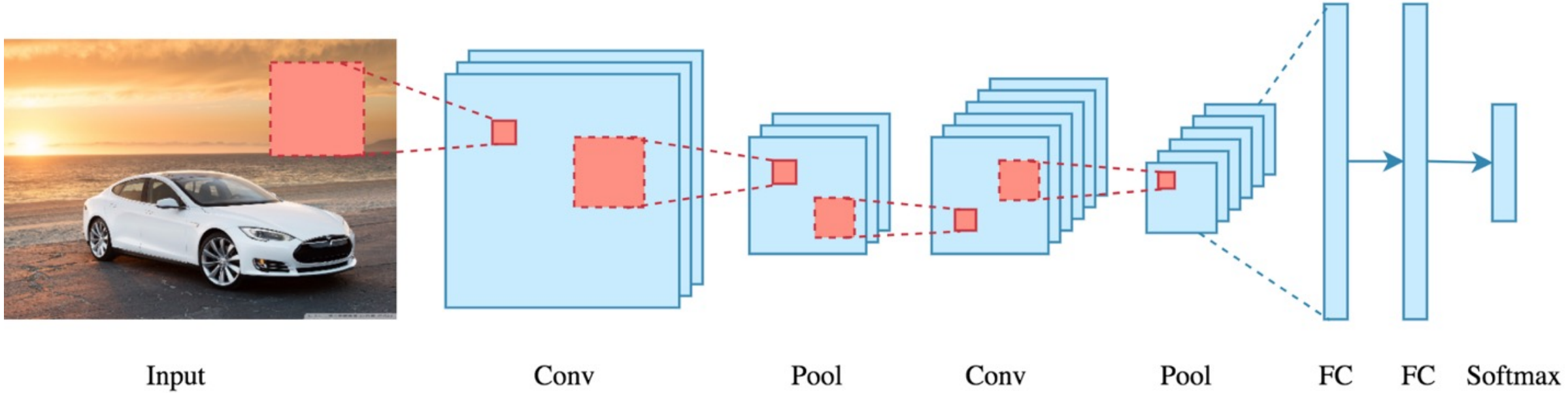
Source: 3Blue1Brown (2017), What is backpropagation really doing? | Chapter 3, deep learning, https://www.youtube.com/watch?v=Ilg3gGewQ5U

Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN)

- **Convolution**
- **Pooling**
- **Fully Connection (FC) (Flattening)**

CNN Architecture



Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Convolution Layer

Convolution is a **mathematical operation** to **merge two sets** of information

3x3 convolution

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Convolution Layer
Input x Filter --> Feature Map

receptive field: 3x3

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

Input x Filter

4		

Feature Map

Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Convolution Layer
Input x Filter --> Feature Map

receptive field: 3x3

1	1x1	1x0	0x1	0
0	1x0	1x1	1x0	0
0	0x1	1x0	1x1	1
0	0	1	1	0
0	1	1	0	0

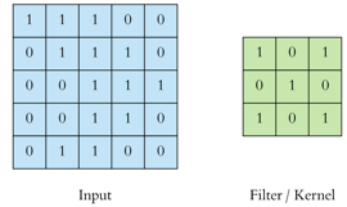
Input x Filter

4	3	

Feature Map

Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Convolution Layer



1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

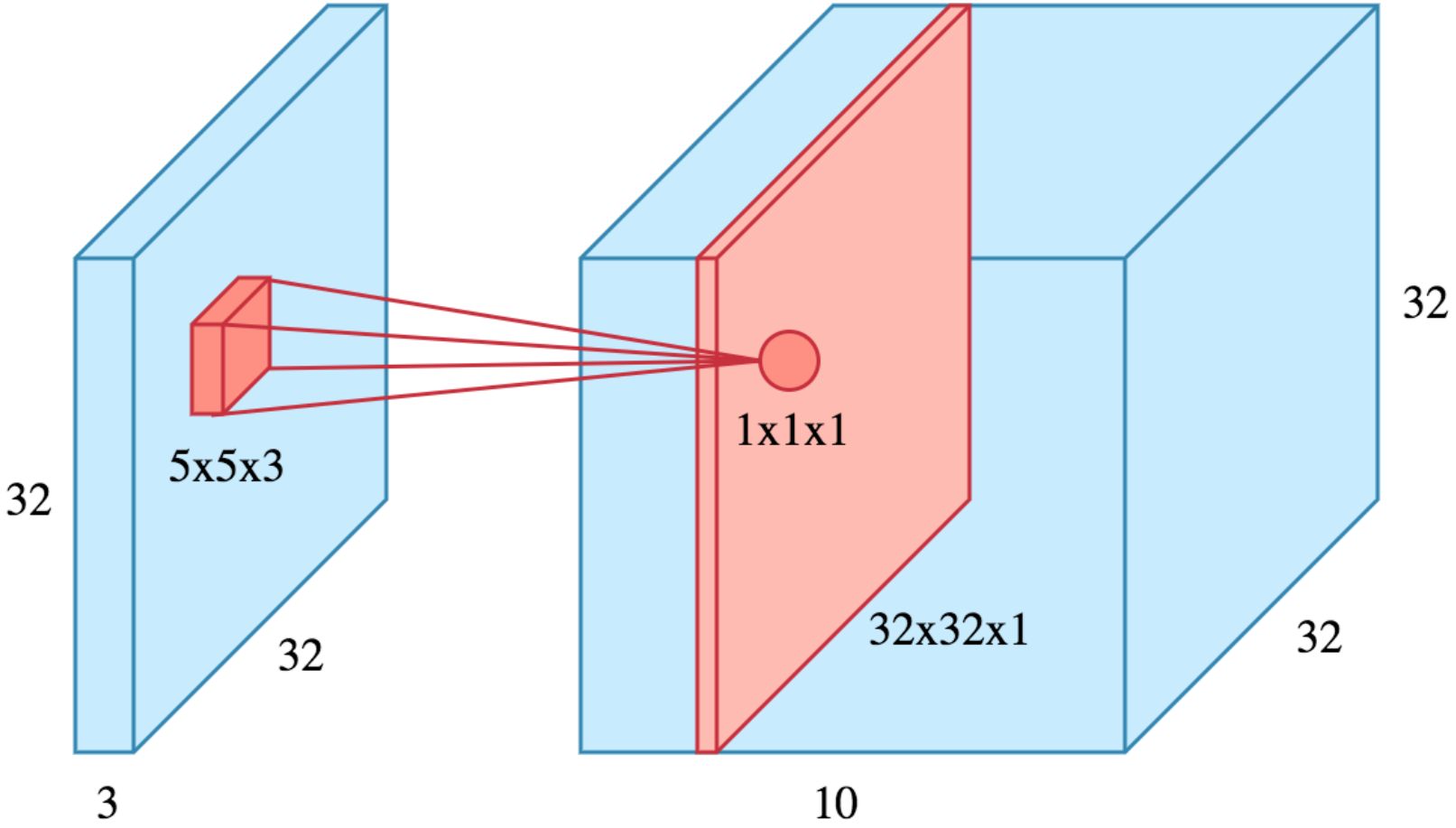
4		

Example convolution operation shown in 2D using a 3x3 filter

Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Convolution Layer

10 different filters 10 feature maps of size 32x32x1

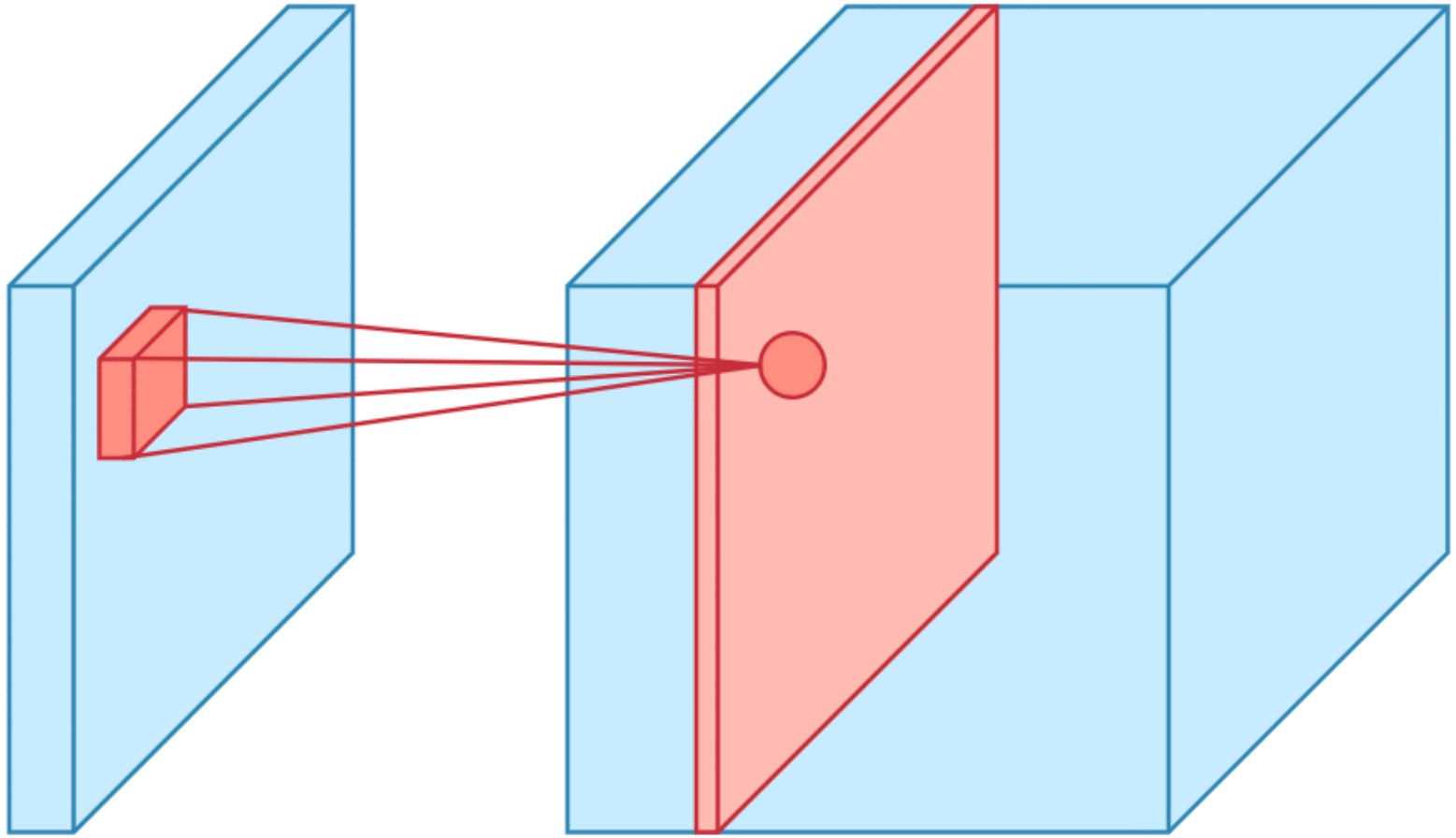


final output of the convolution layer:
a volume of size 32x32x10

Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

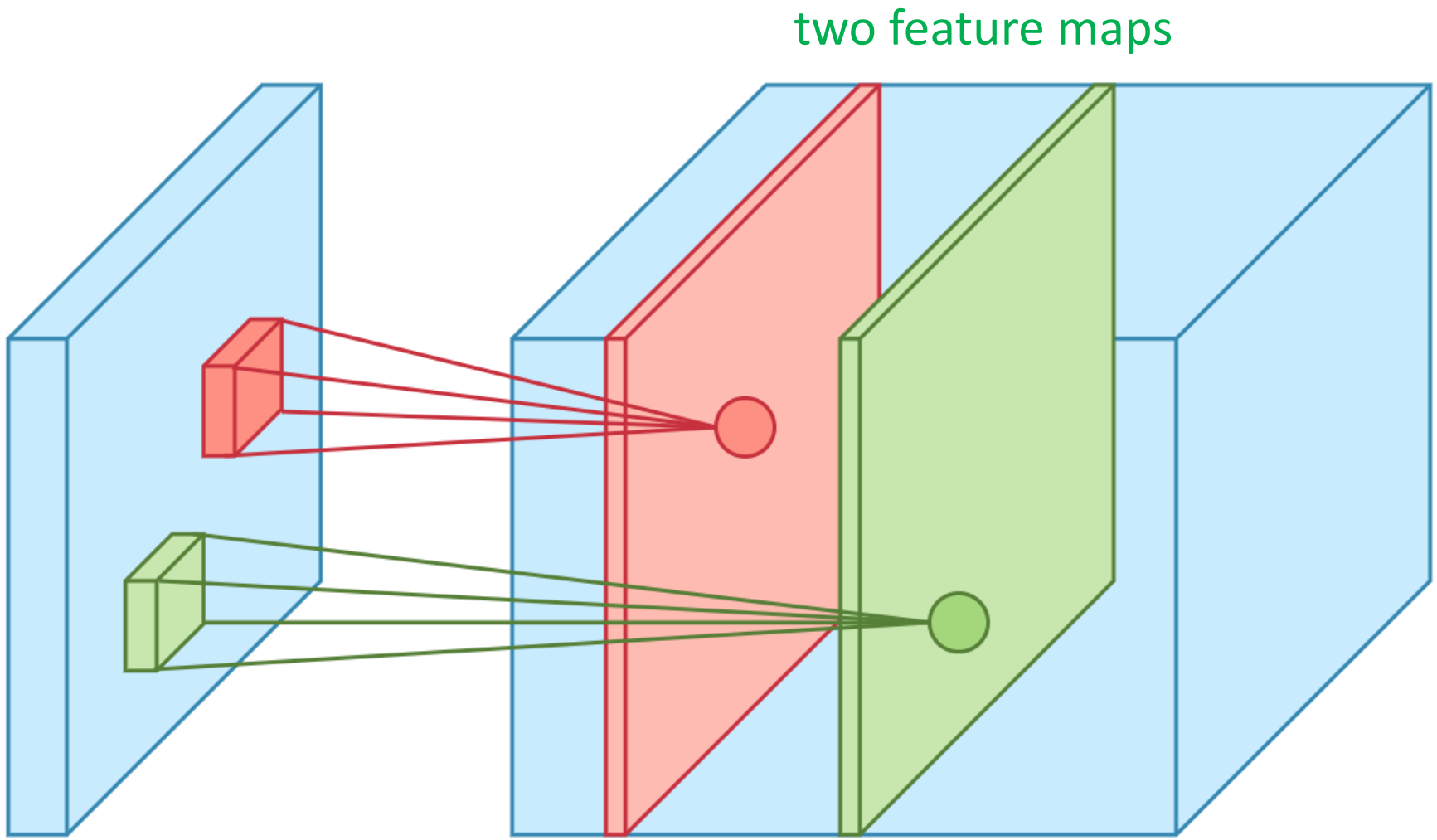
CNN Convolution Layer

Sliding operation at 4 locations



Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

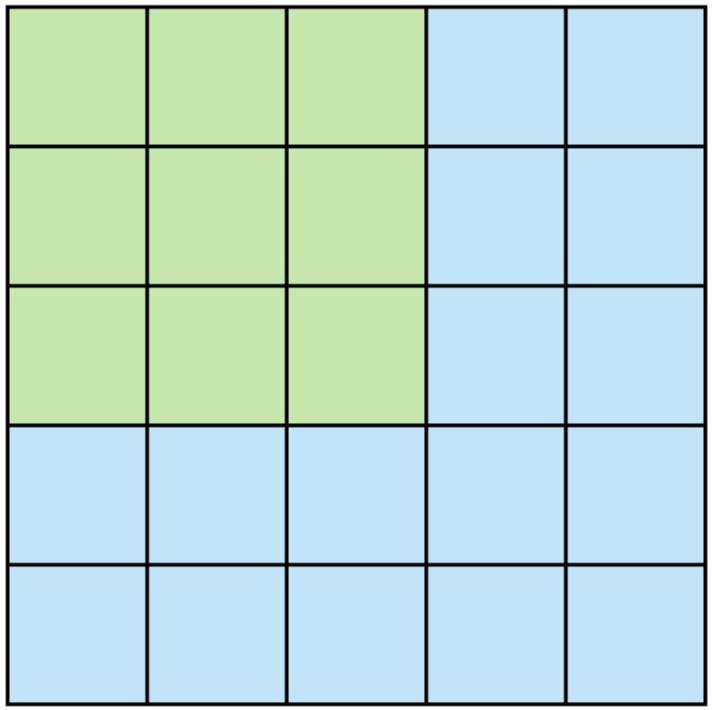
CNN Convolution Layer



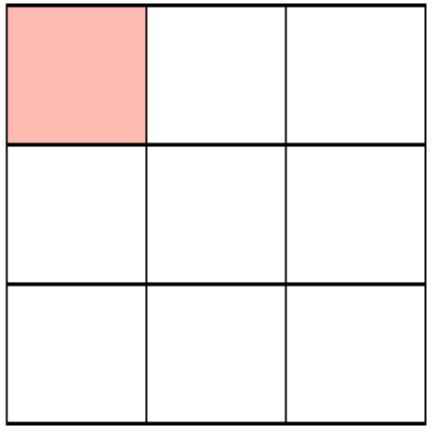
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Convolution Layer

Stride specifies how much
we move the convolution filter at each step



Stride 1

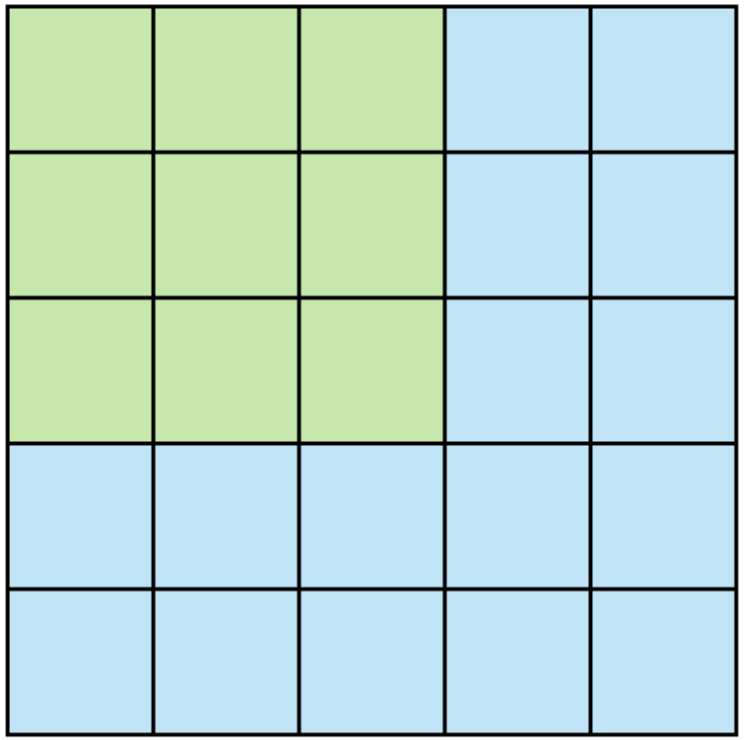


Feature Map

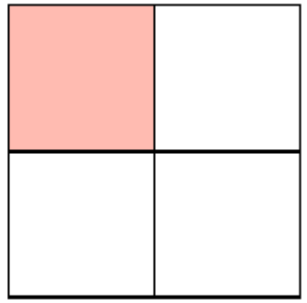
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks,
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Convolution Layer

Stride specifies how much
we move the convolution filter at each step



Stride 2

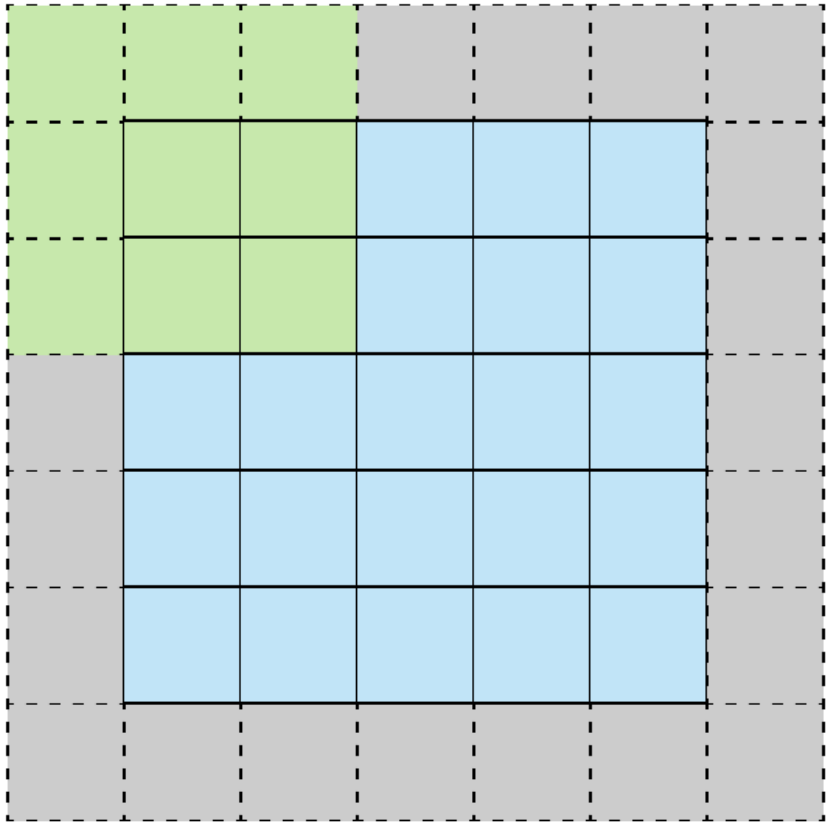


Feature Map

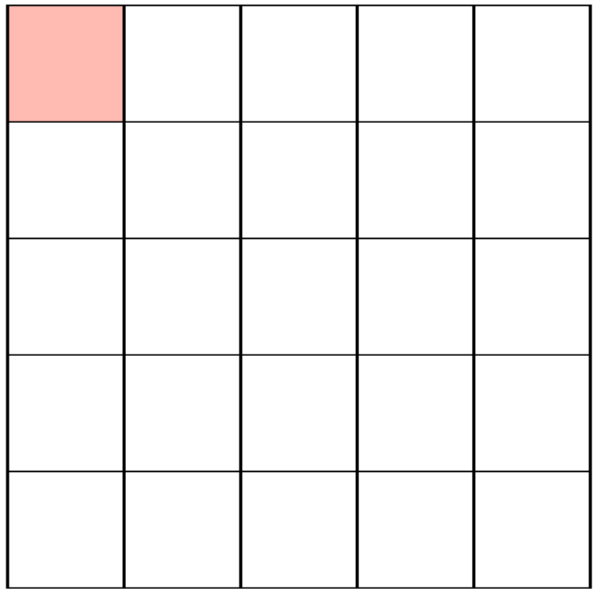
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks,
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Convolution Layer

Stride 1 with Padding



Stride 1 with Padding

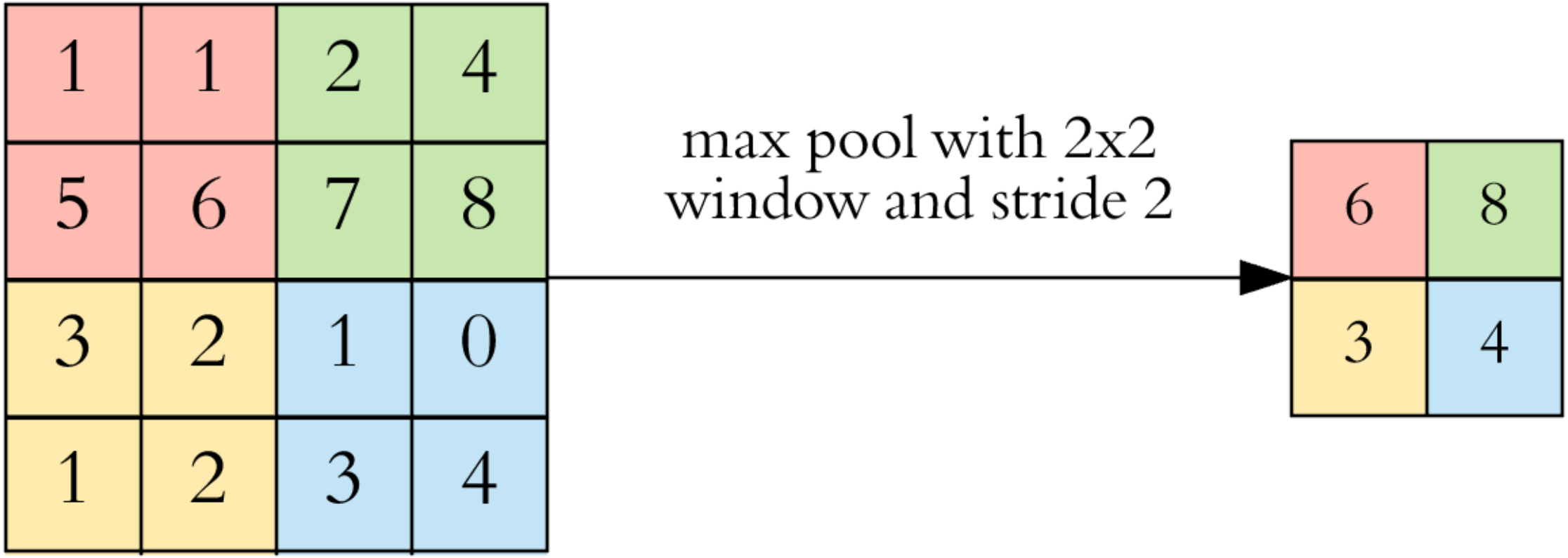


Feature Map

Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

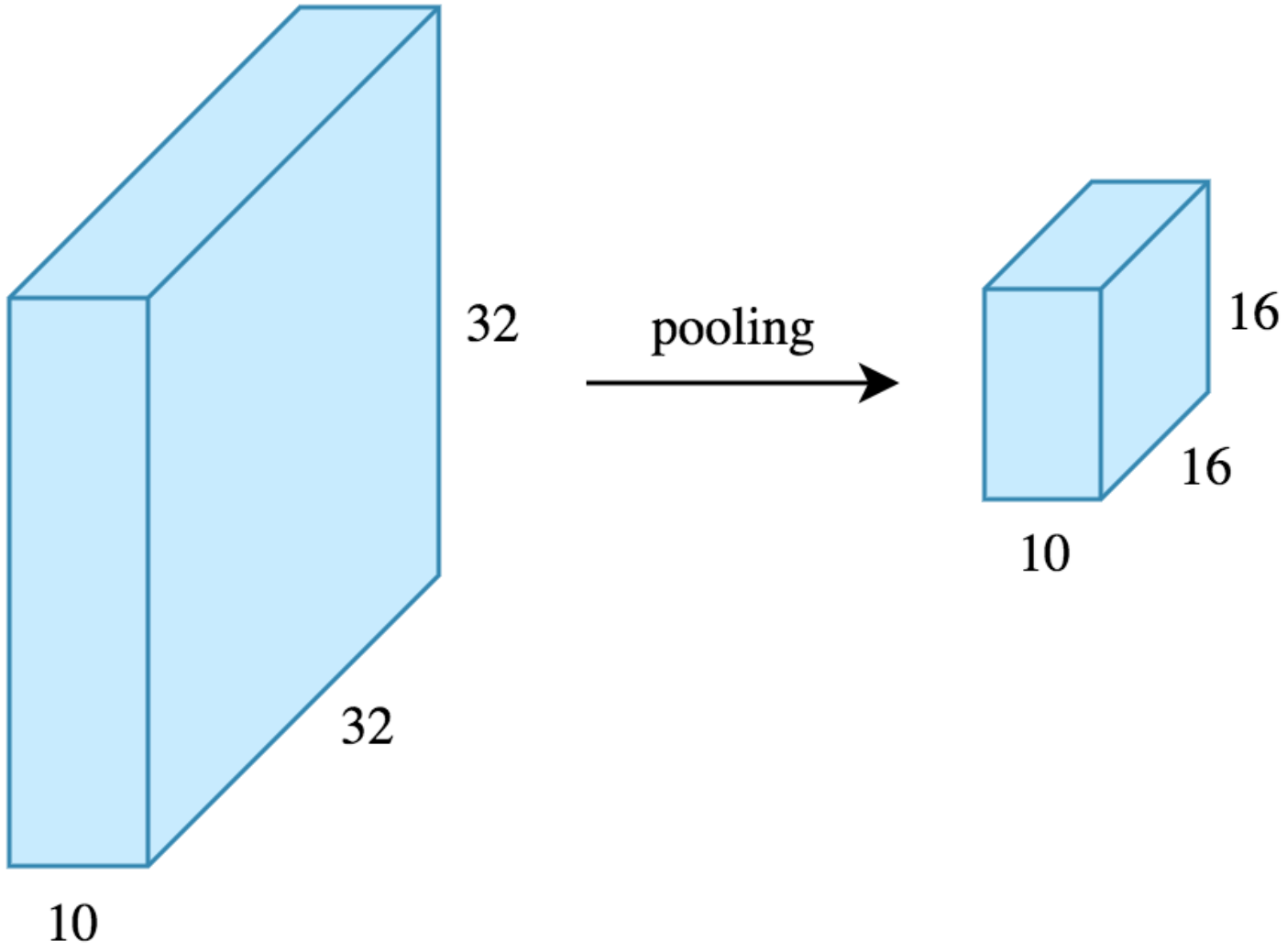
CNN Pooling Layer

Max Pooling



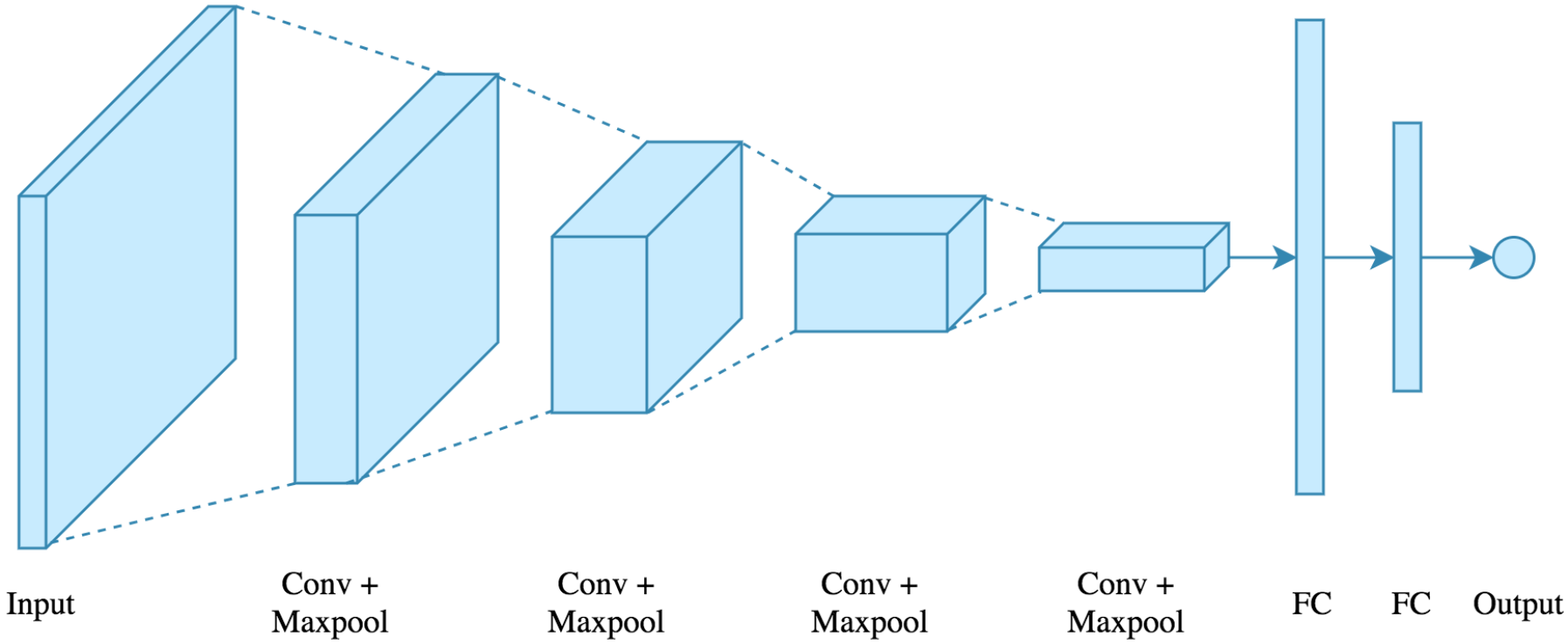
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks,
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Pooling Layer



Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Architecture
4 convolution + pooling layers, followed by 2 fully connected layers



Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

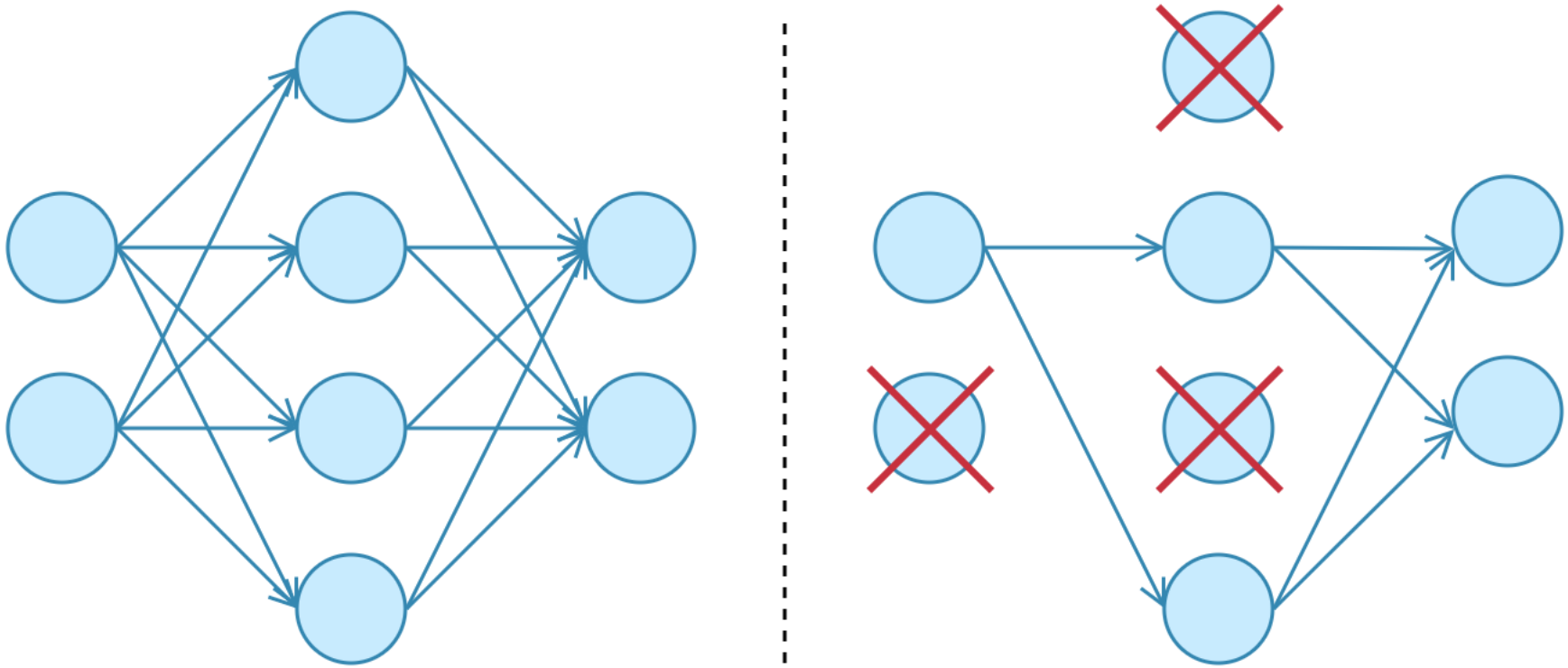
CNN Architecture
4 convolution + pooling layers, followed by 2 fully connected layers

https://gist.github.com/ardendertat/0fc5515057c47e7386fe04e9334504e3

```
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', padding='same', name='conv_1',
                 input_shape=(150, 150, 3)))
model.add(MaxPooling2D((2, 2), name='maxpool_1'))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same', name='conv_2'))
model.add(MaxPooling2D((2, 2), name='maxpool_2'))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same', name='conv_3'))
model.add(MaxPooling2D((2, 2), name='maxpool_3'))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same', name='conv_4'))
model.add(MaxPooling2D((2, 2), name='maxpool_4'))
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(512, activation='relu', name='dense_1'))
model.add(Dense(128, activation='relu', name='dense_2'))
model.add(Dense(1, activation='sigmoid', name='output'))
```

Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Dropout

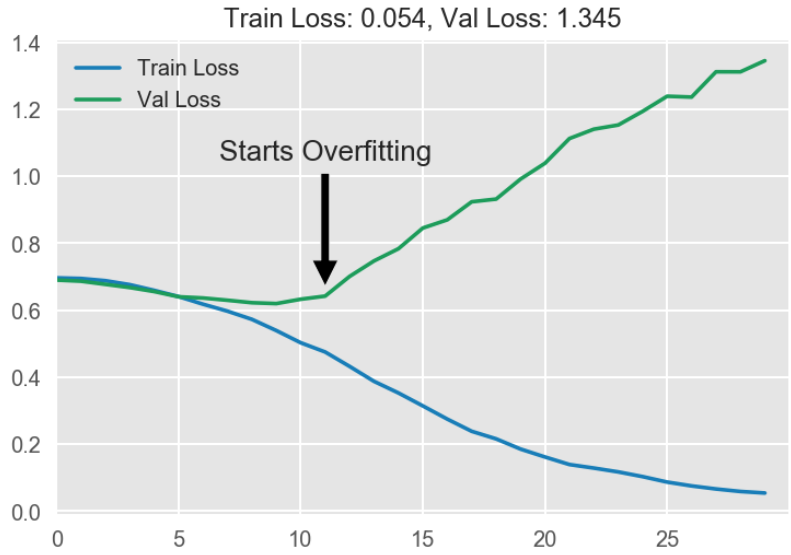


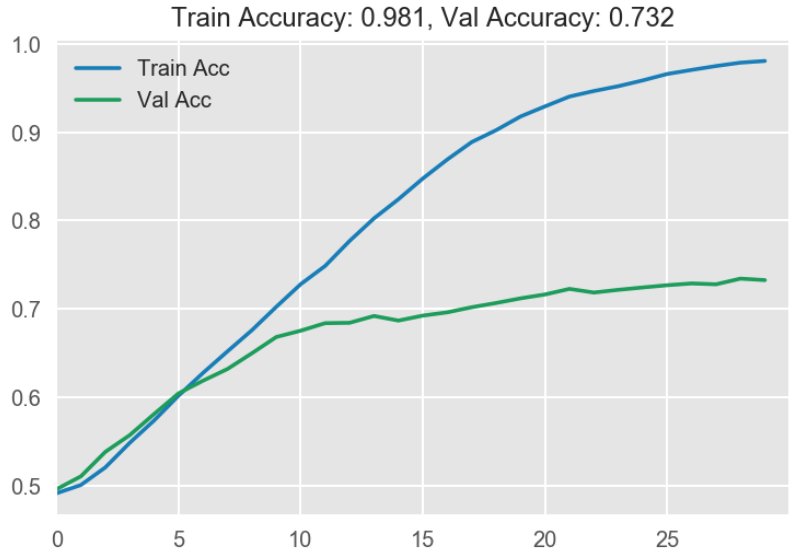
No Dropout

With Dropout

Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Model Performance

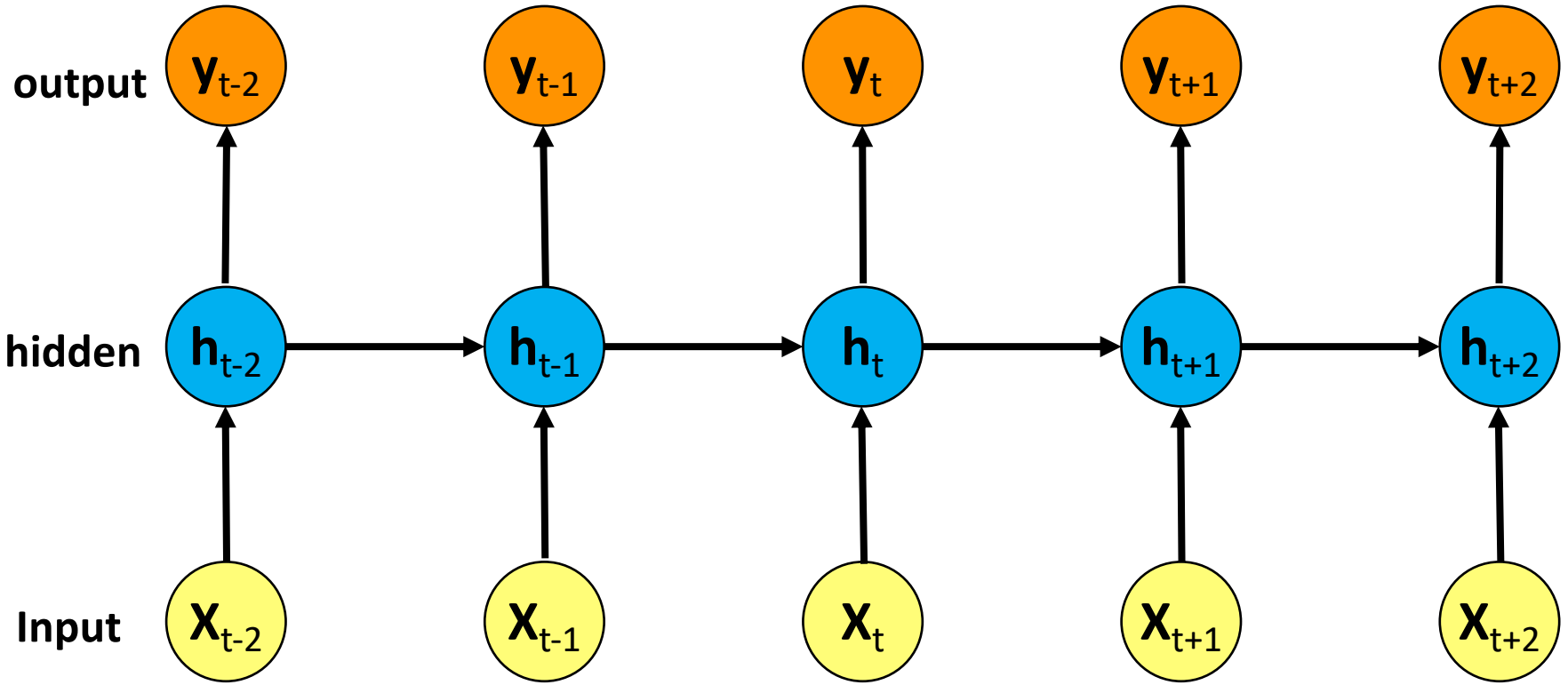




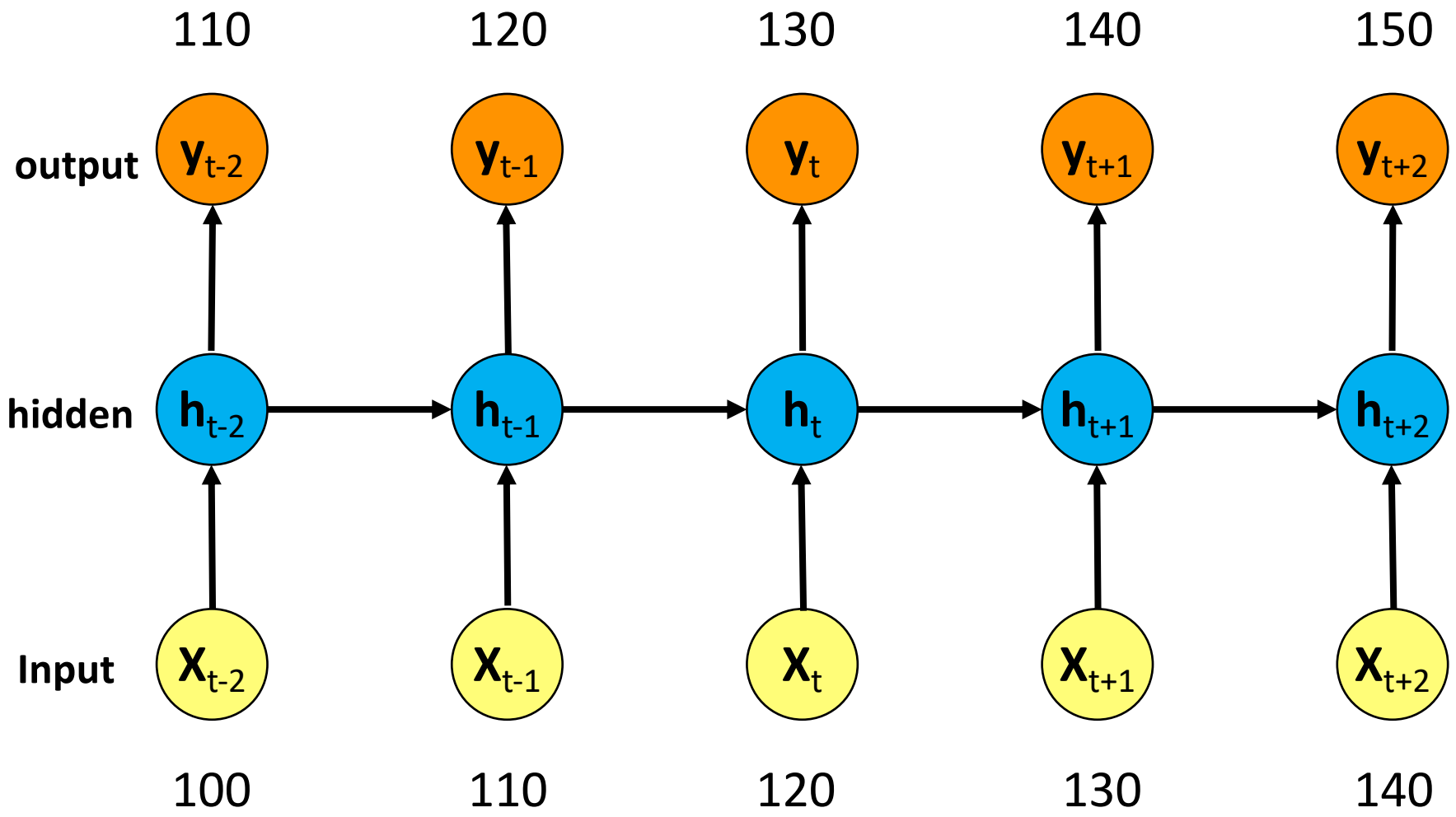
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Recurrent Neural Networks (RNN)

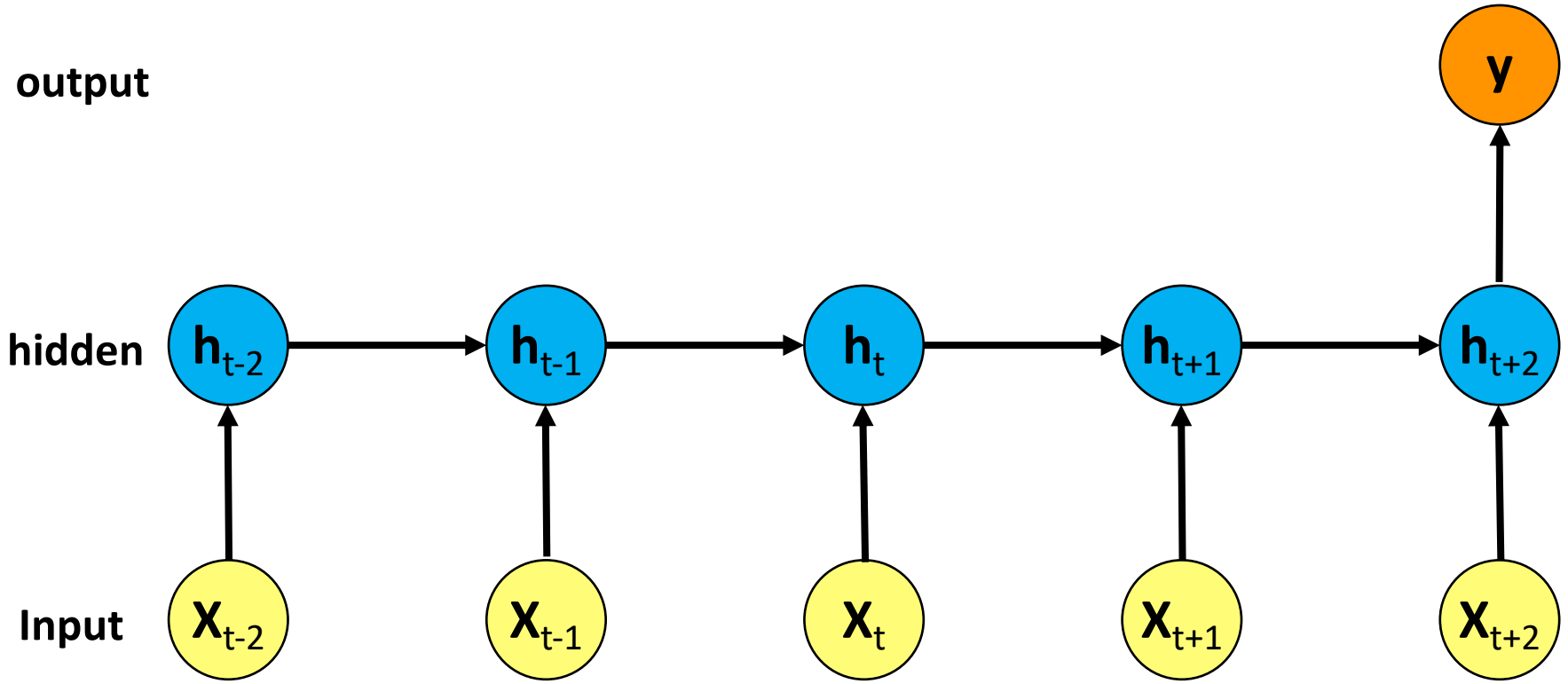
Recurrent Neural Networks (RNN)



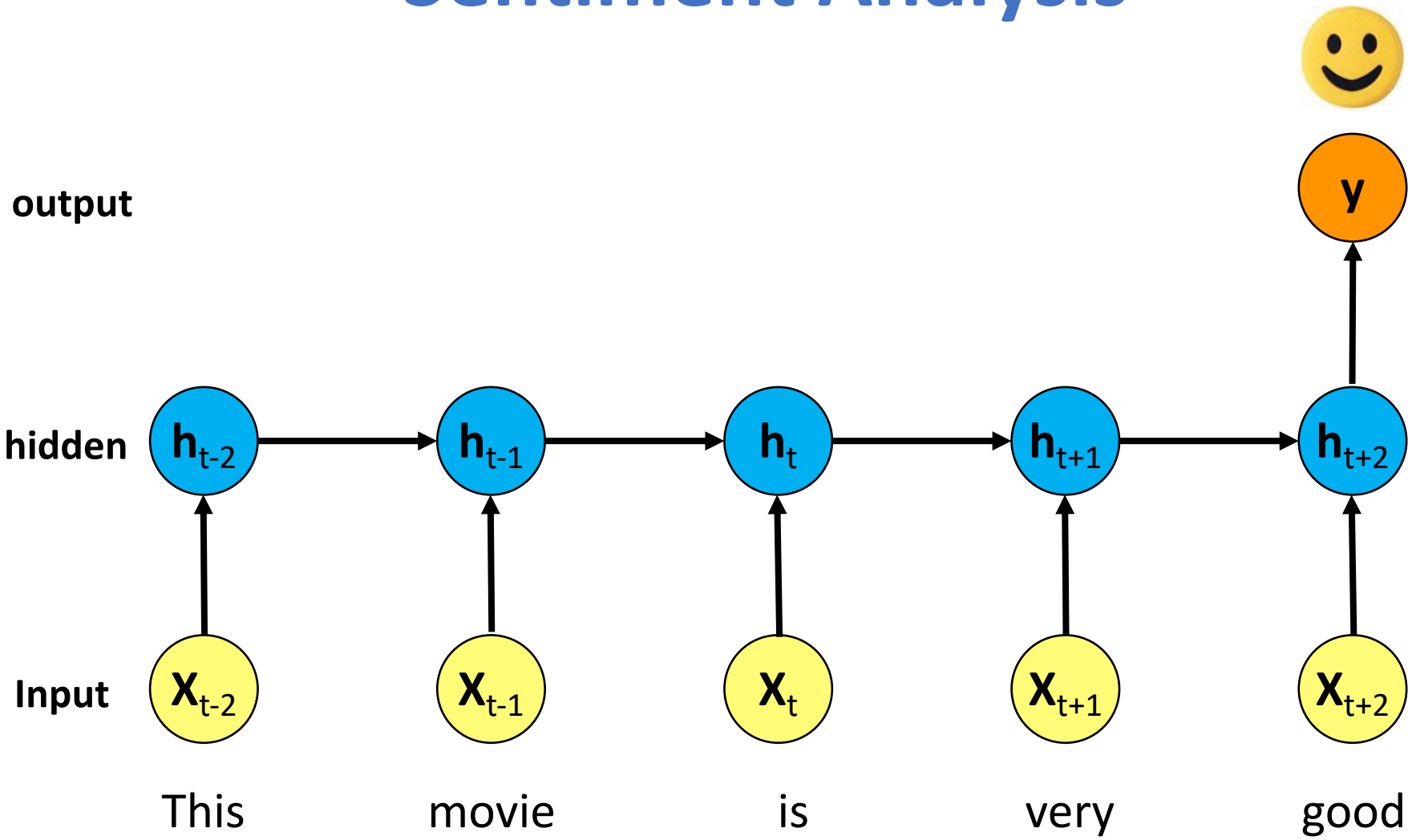
Recurrent Neural Networks (RNN)
Time Series Forecasting



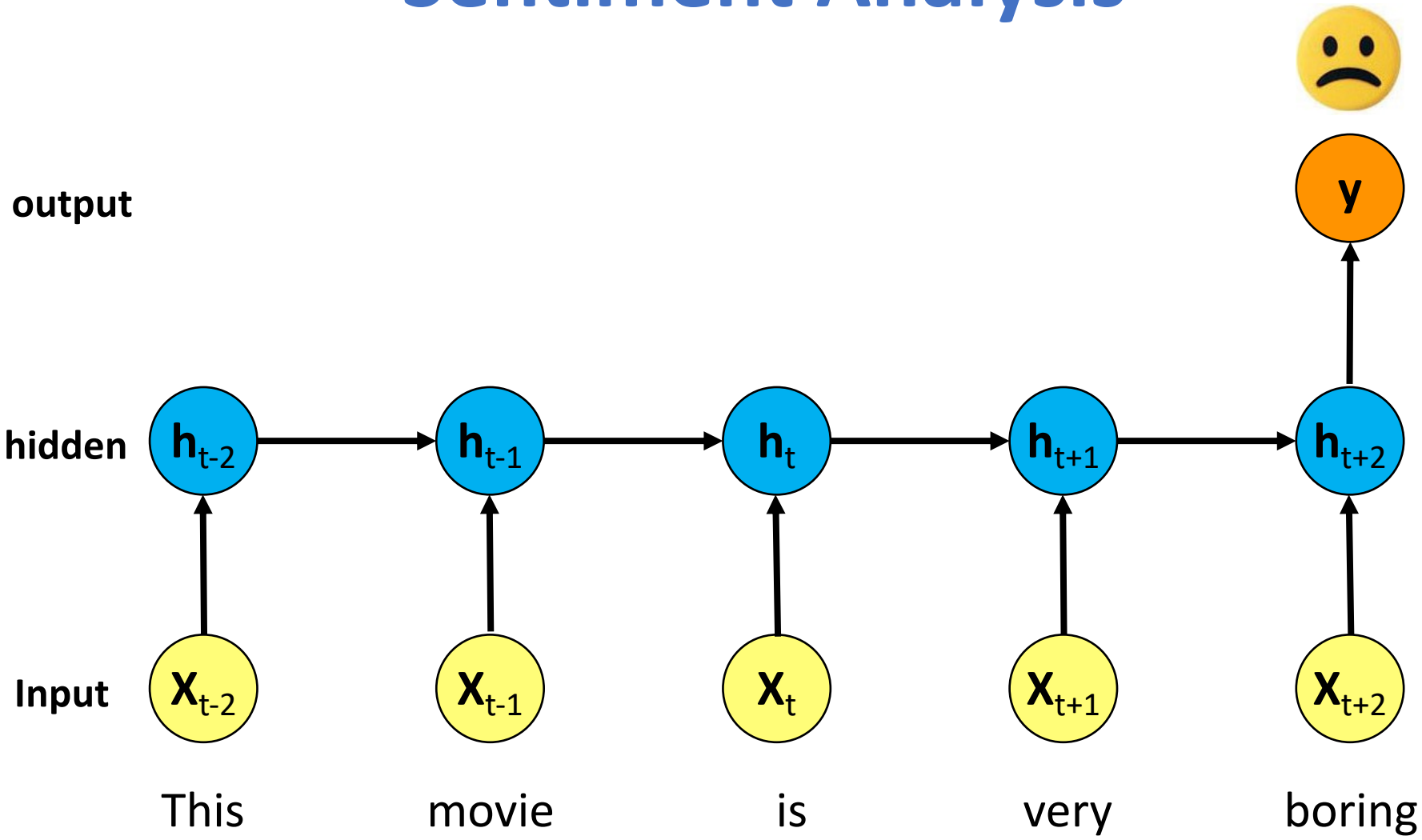
Recurrent Neural Networks (RNN)



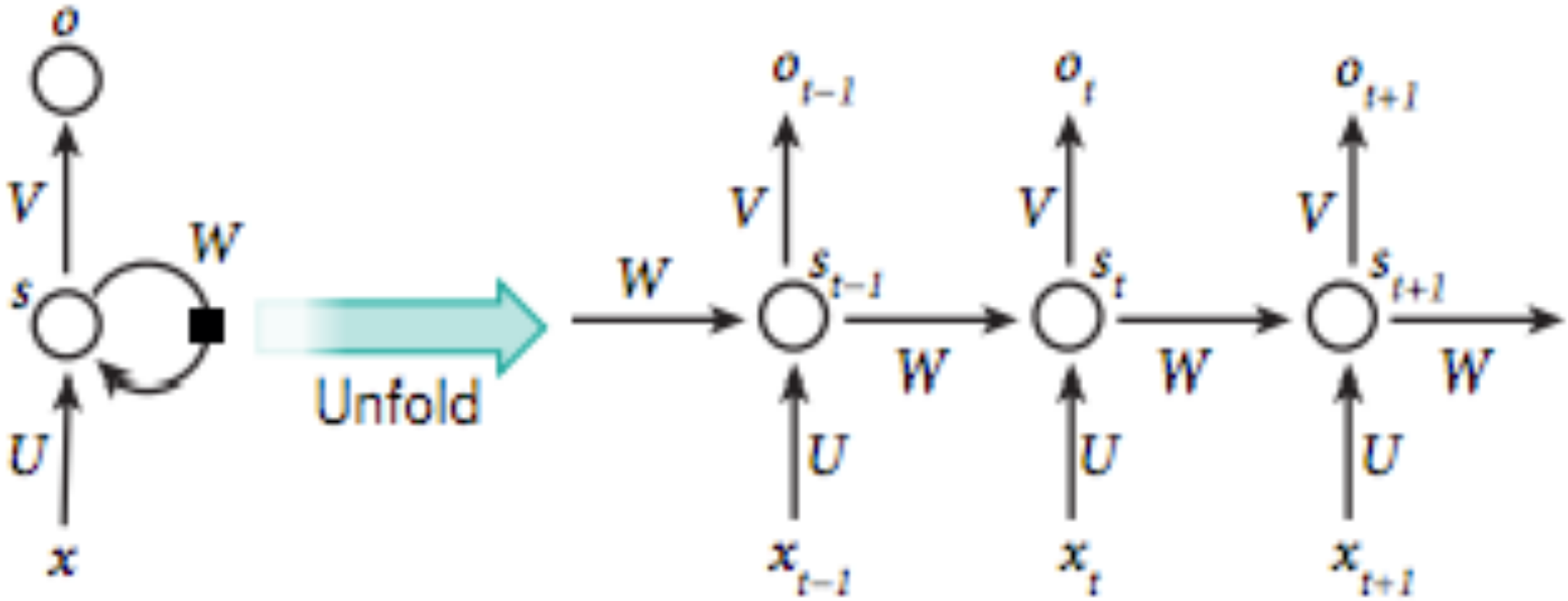
Recurrent Neural Networks (RNN)
Sentiment Analysis



Recurrent Neural Networks (RNN)
Sentiment Analysis

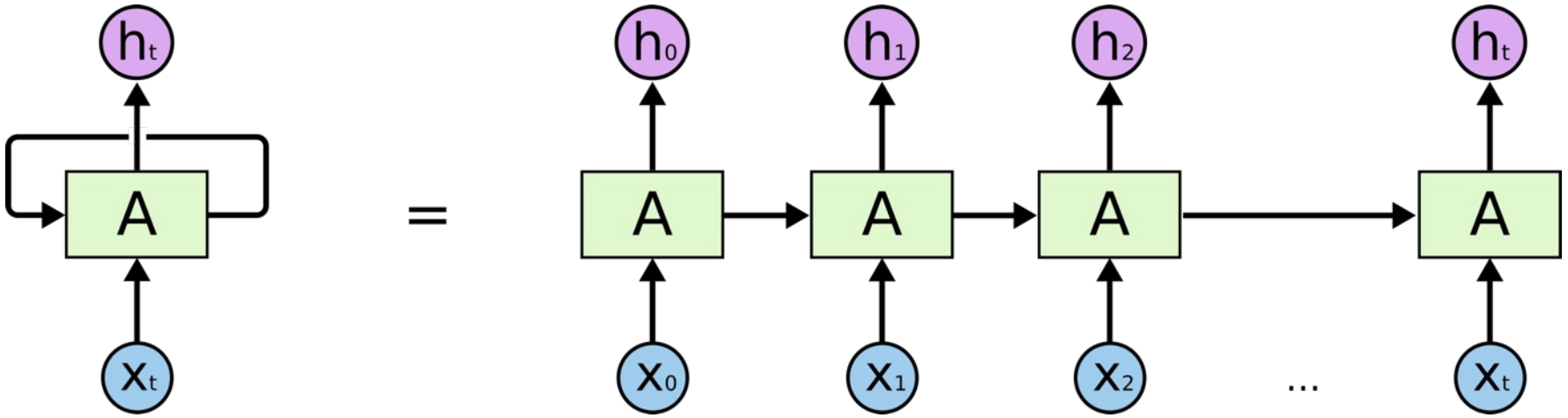


Recurrent Neural Network (RNN)



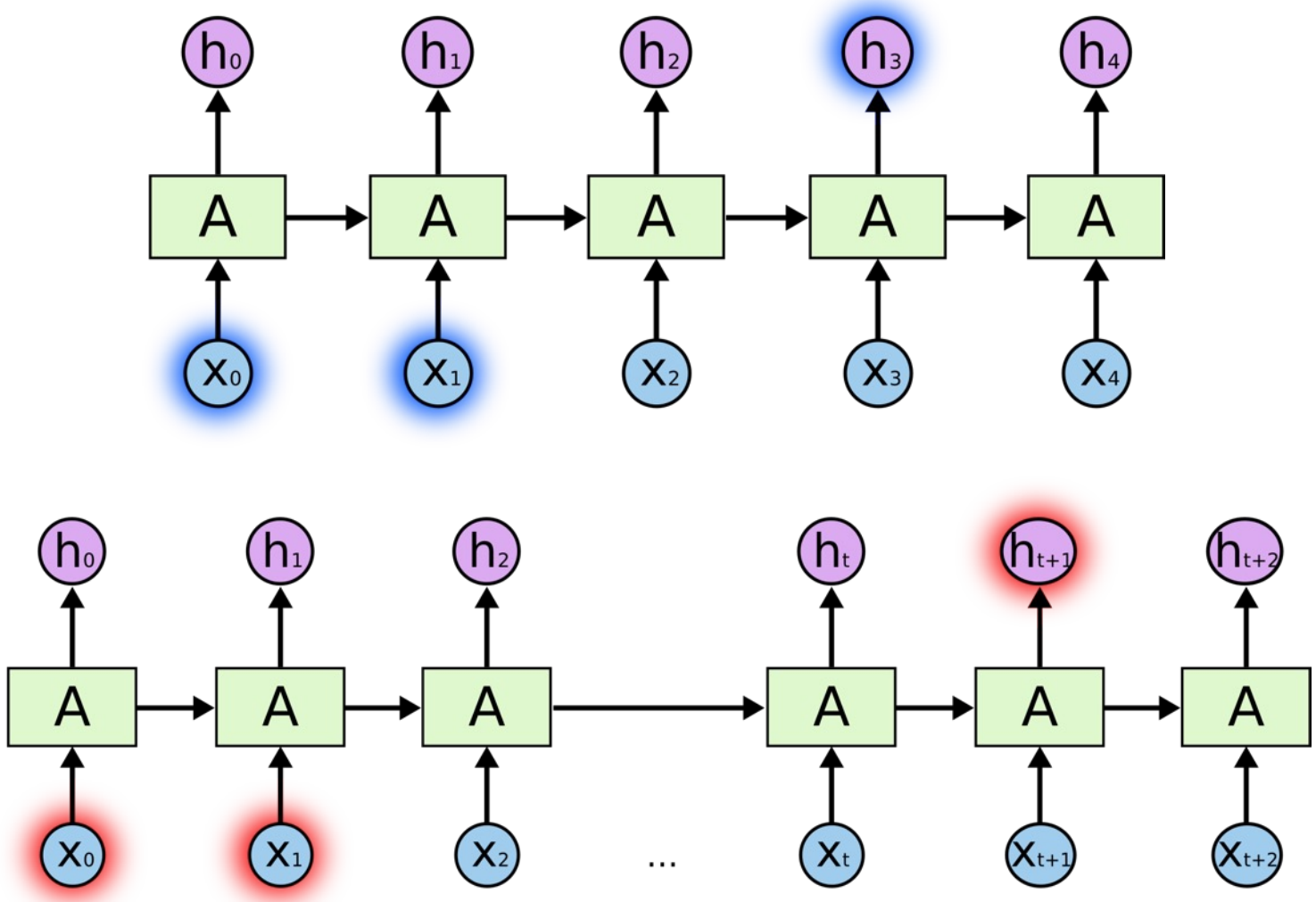
Source: LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521, no. 7553 (2015): 436-444.

RNN



Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

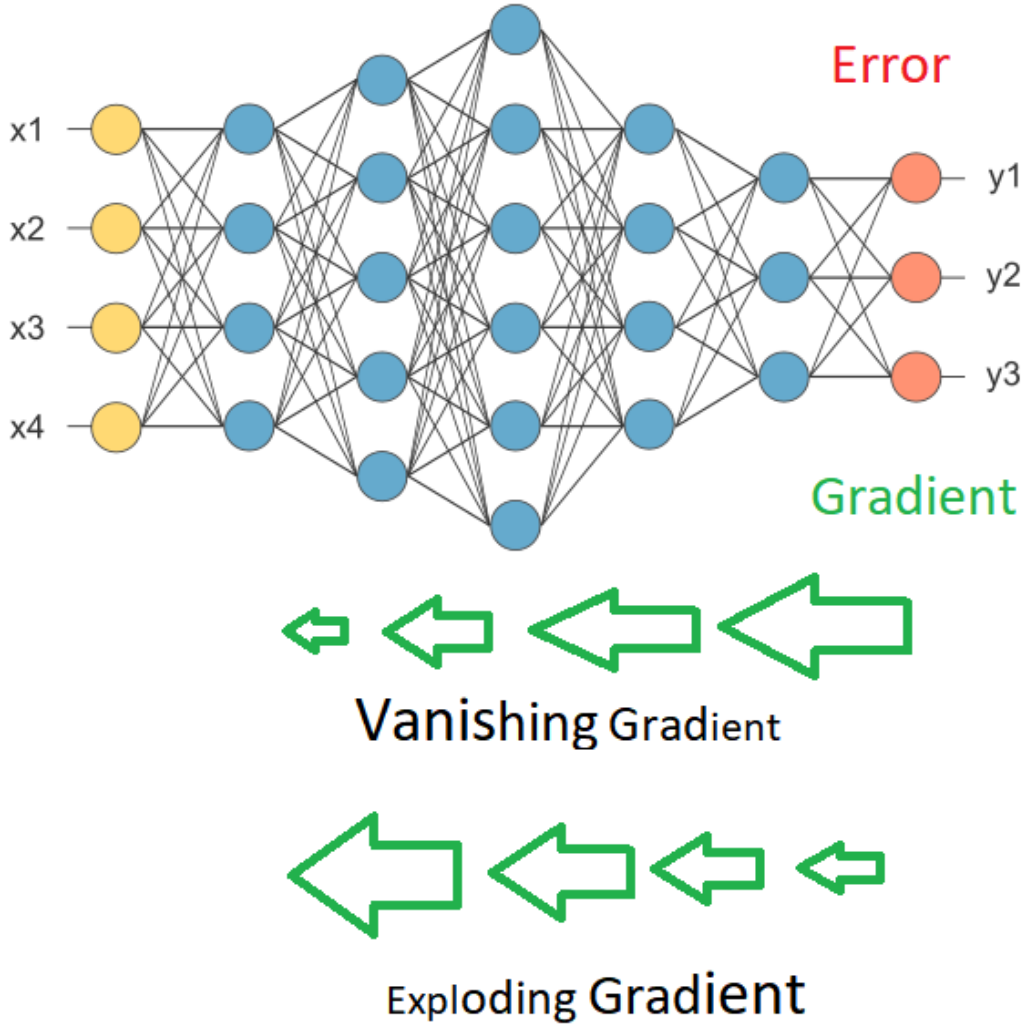
RNN long-term dependencies



I grew up in France… I speak fluent French.

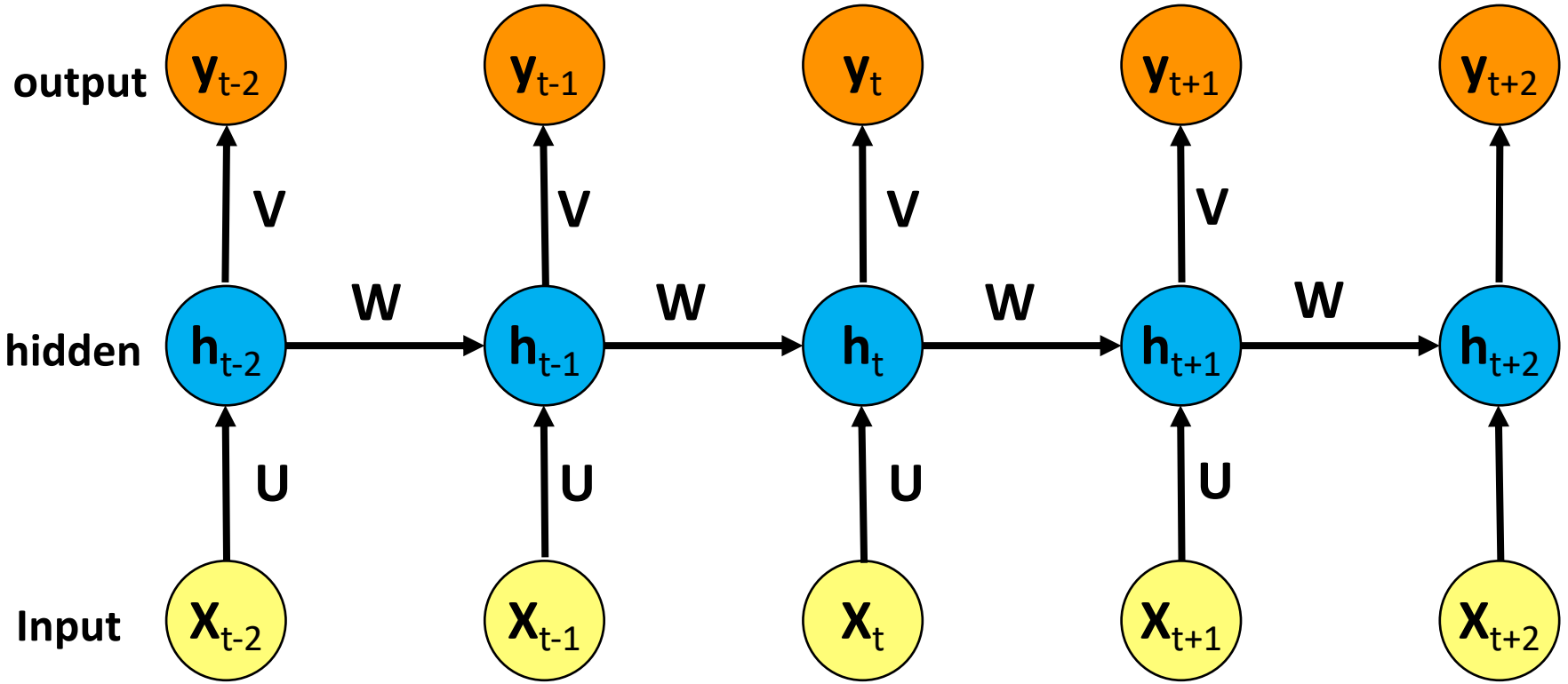
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Vanishing Gradient
Exploding Gradient

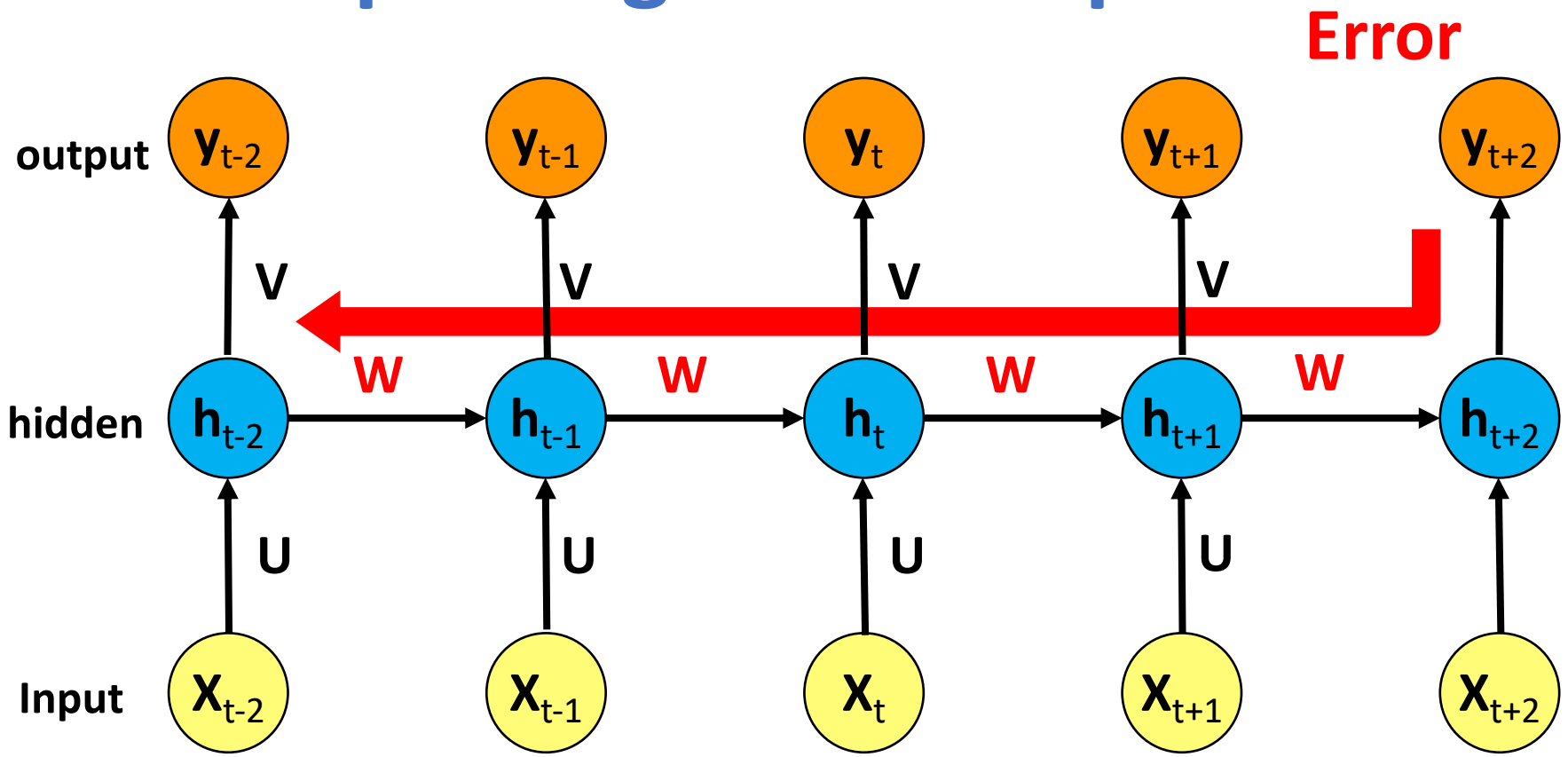


Source: https://medium.com/deep-math-machine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235

Recurrent Neural Networks (RNN)



RNN
Vanishing Gradient problem
Exploding Gradient problem

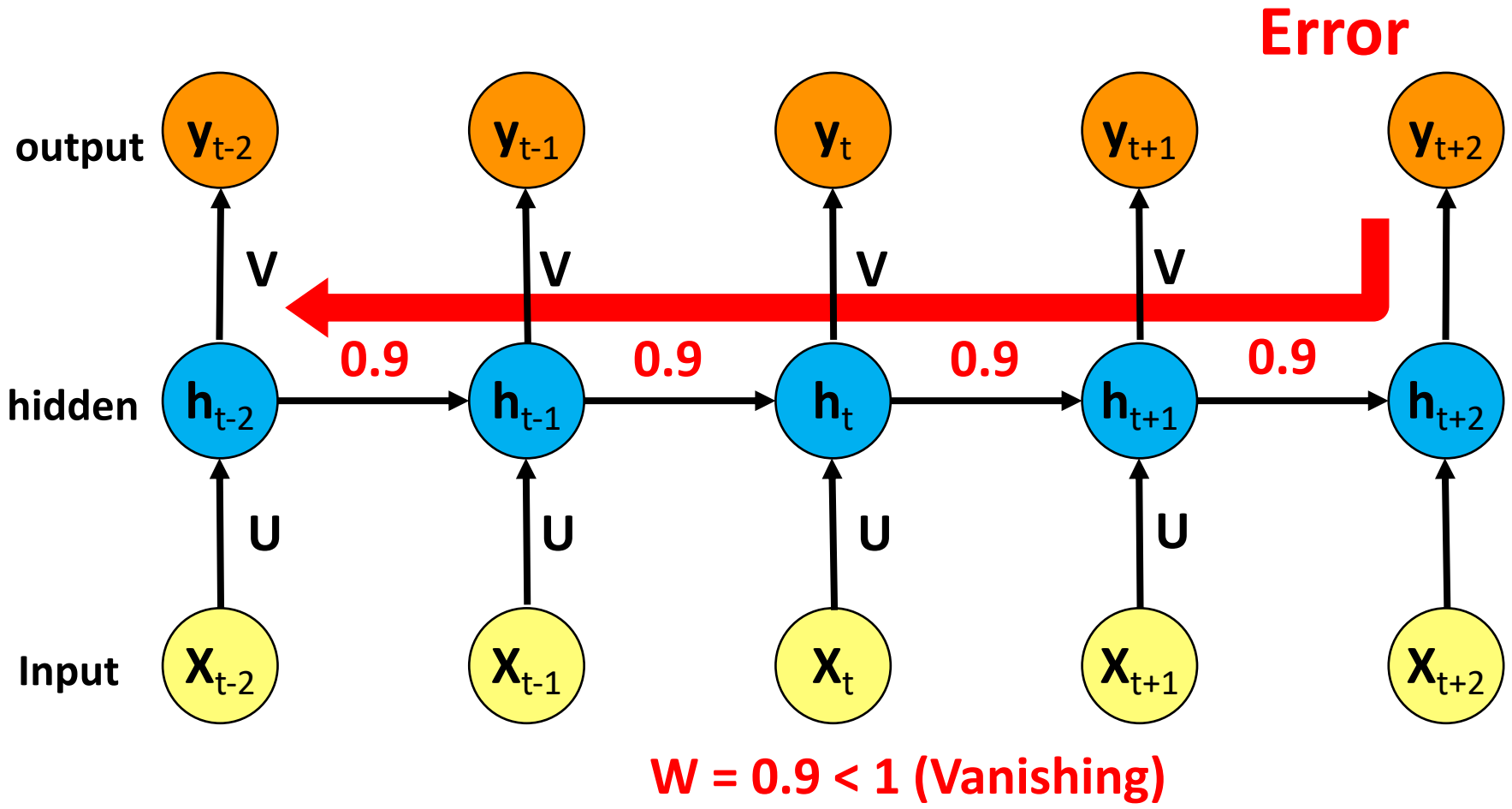


if |W| < 1 (Vanishing)

if |W| > 1 (Exploding)

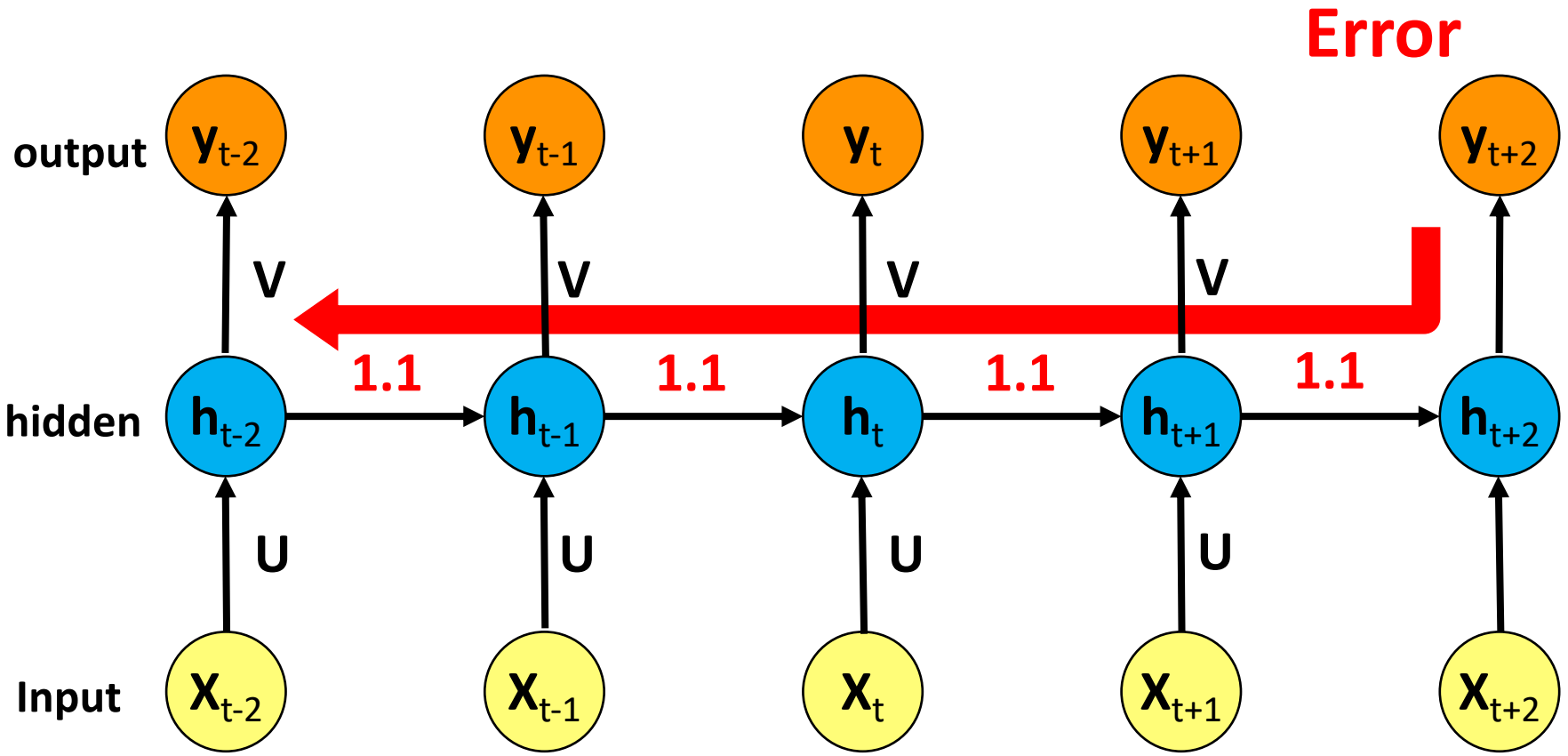
Source: https://medium.com/deep-math-machine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235

RNN
Vanishing Gradient problem



Source: https://medium.com/deep-math-machine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235

RNN
Exploding Gradient problem

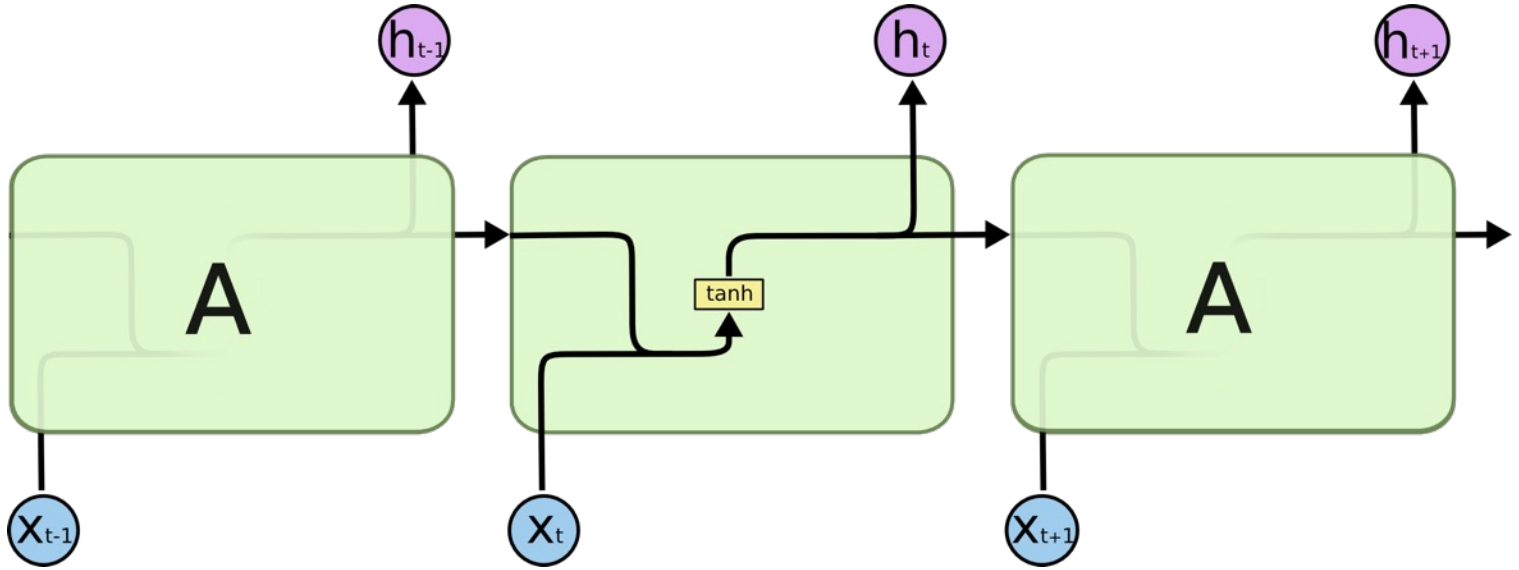


W = 1.1 > 1 (Exploding)

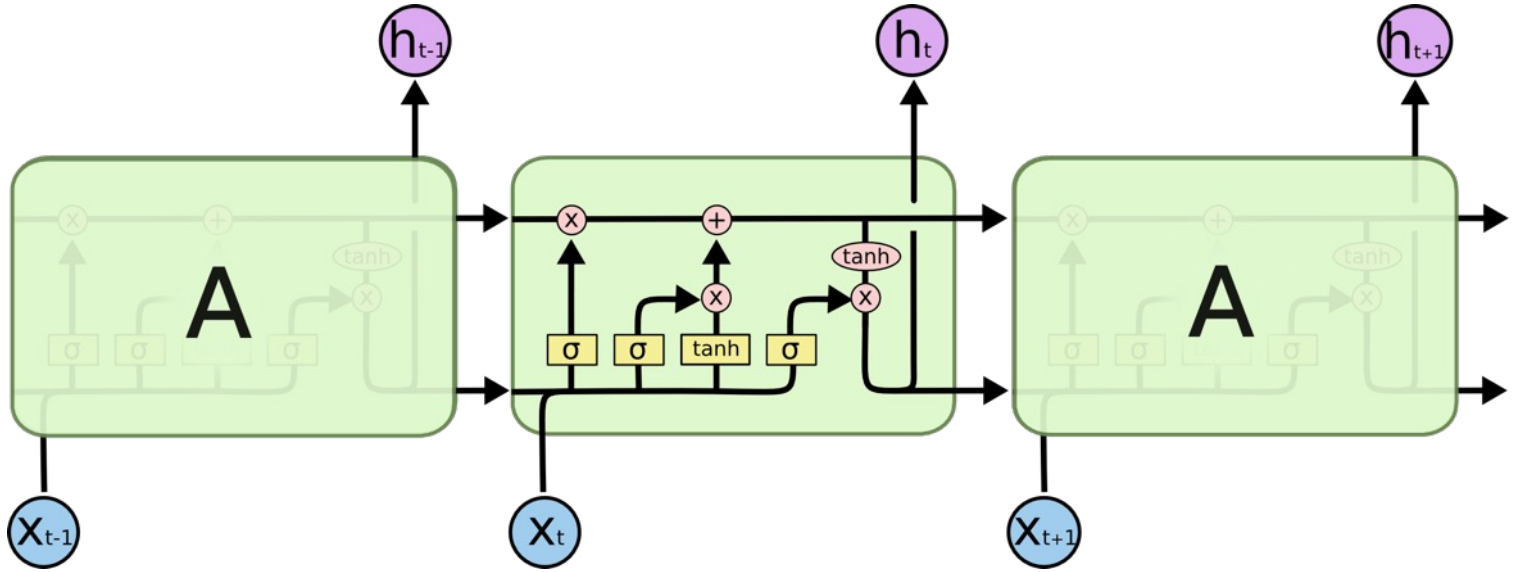
Source: https://medium.com/deep-math-machine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235

RNN LSTM

RNN

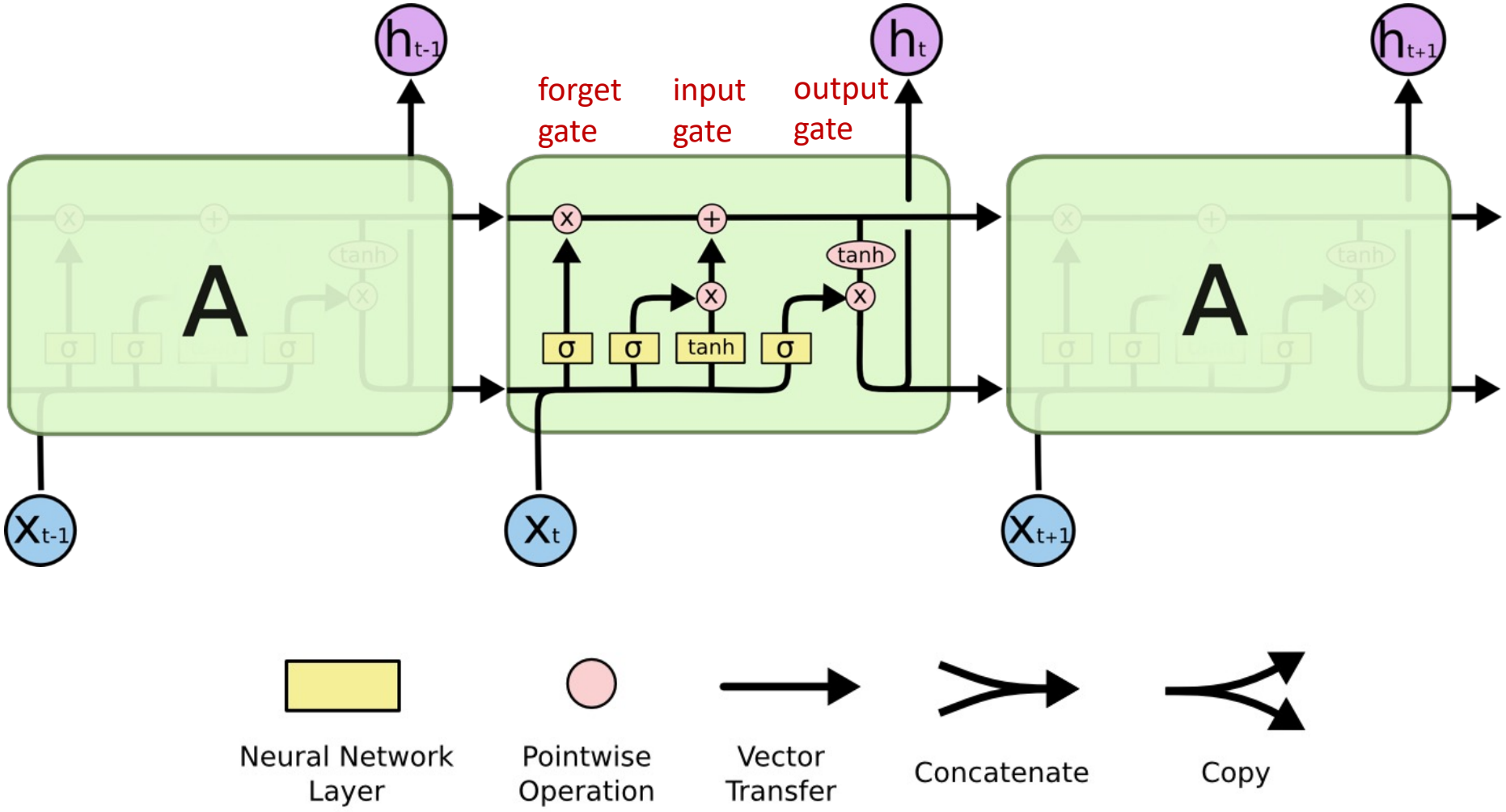


LSTM



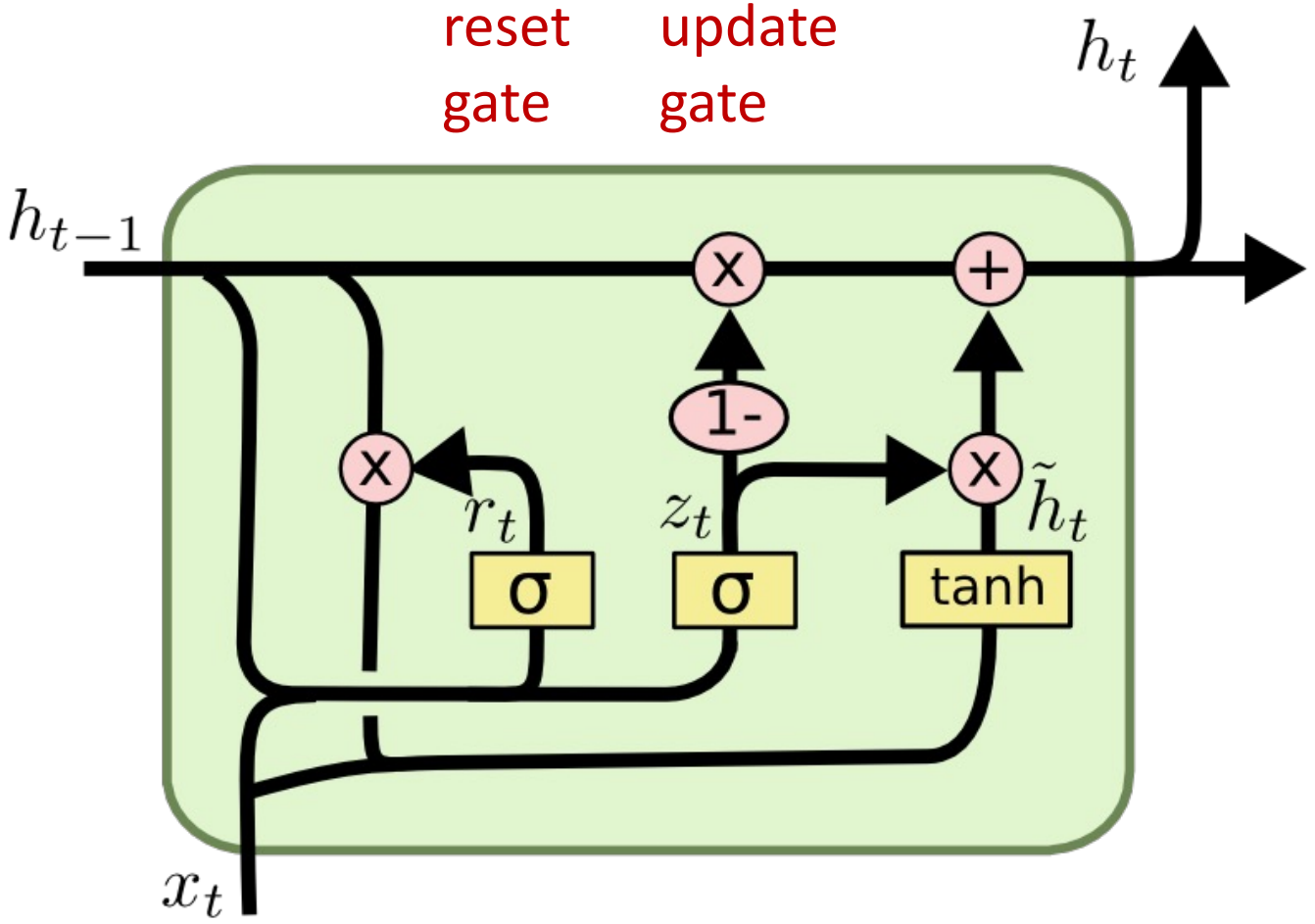
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory (LSTM)



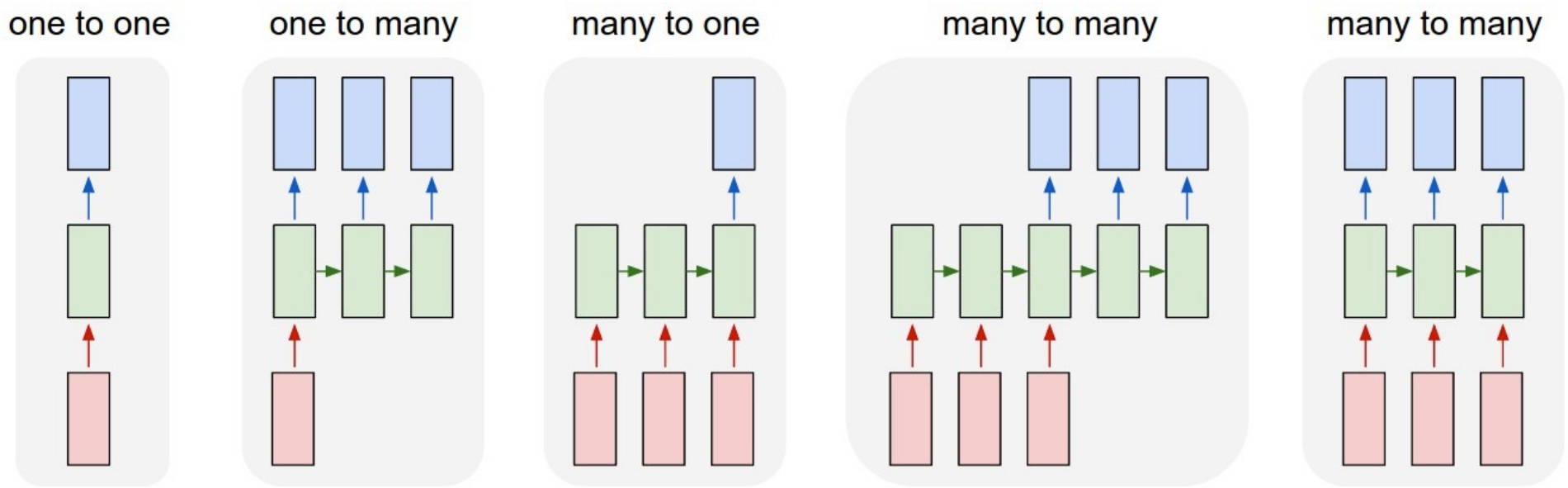
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Gated Recurrent Unit (GRU)



Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM Recurrent Neural Network



Traditional Neural Network

Music Generation

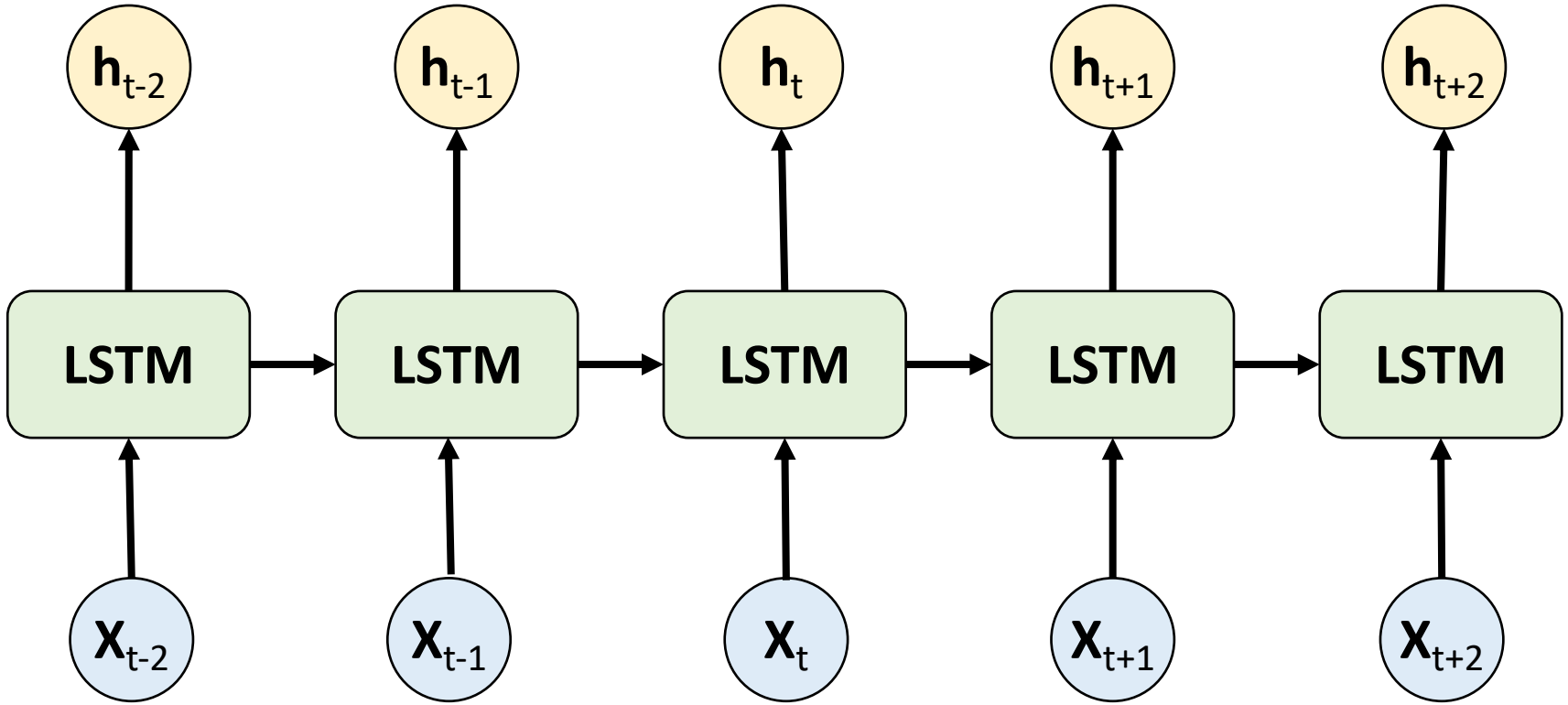
Sentiment Classification

Name Entity Recognition

Machine Translation

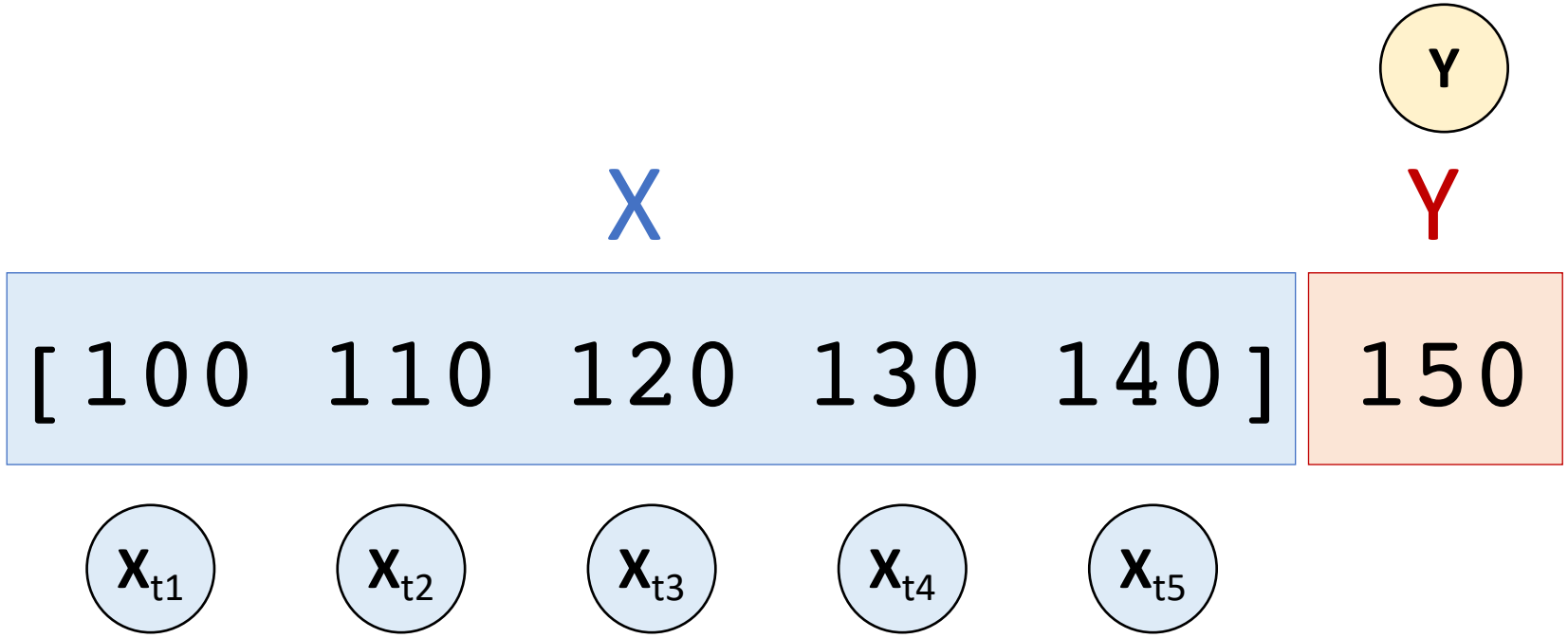
Source: https://github.com/Vict0rSch/deep_learning/tree/master/keras/recurrent

Long Short Term Memory (LSTM) for Time Series Forecasting



Time Series Data

```
[100, 110, 120, 130, 140, 150]
```



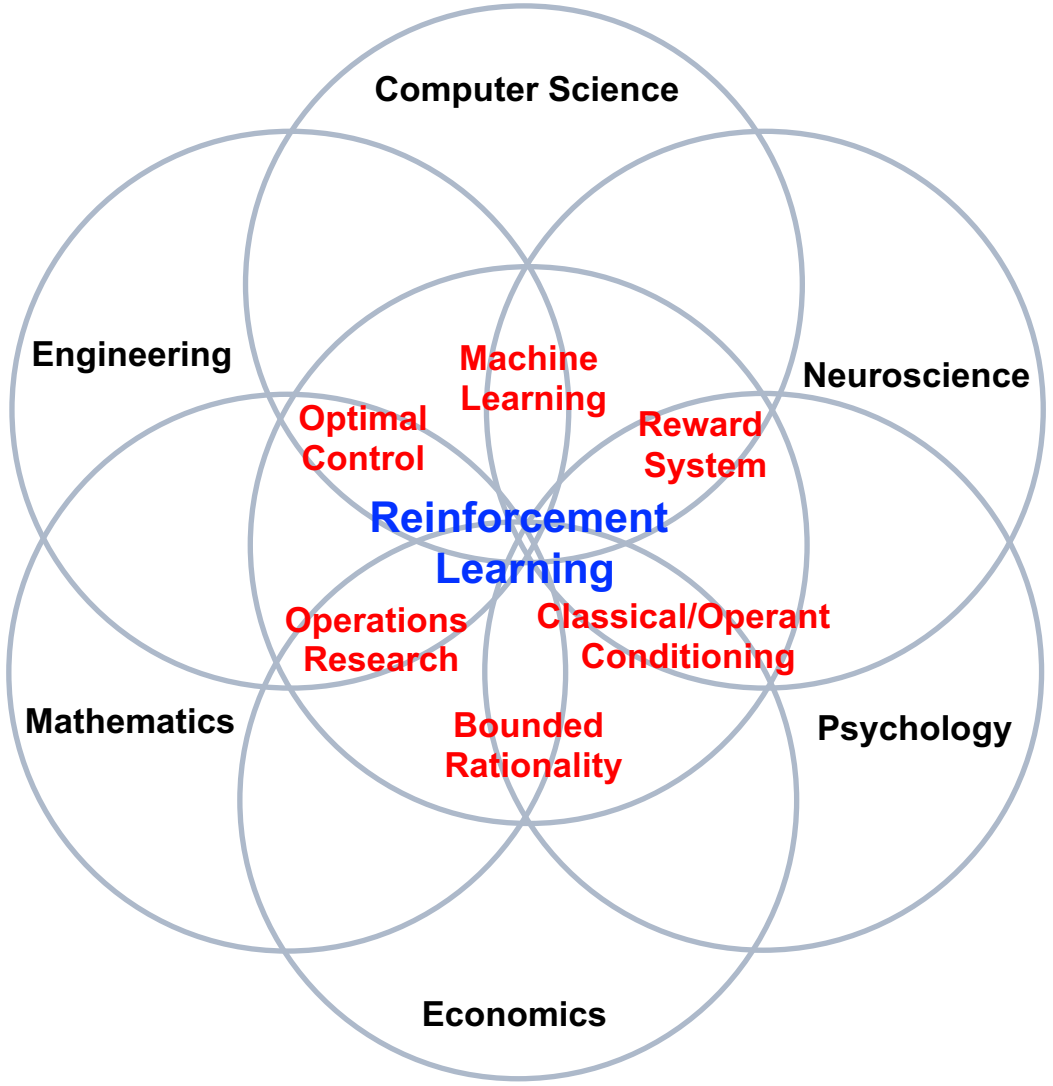
Time Series Data

[10, 20, 30, 40, 50, 60, 70, 80, 90]

X	Y
[10 20 30]	40
[20 30 40]	50
[30 40 50]	60
[40 50 60]	70
[50 60 70]	80
[60 70 80]	90

Reinforcement Learning

Reinforcement Learning (RL)

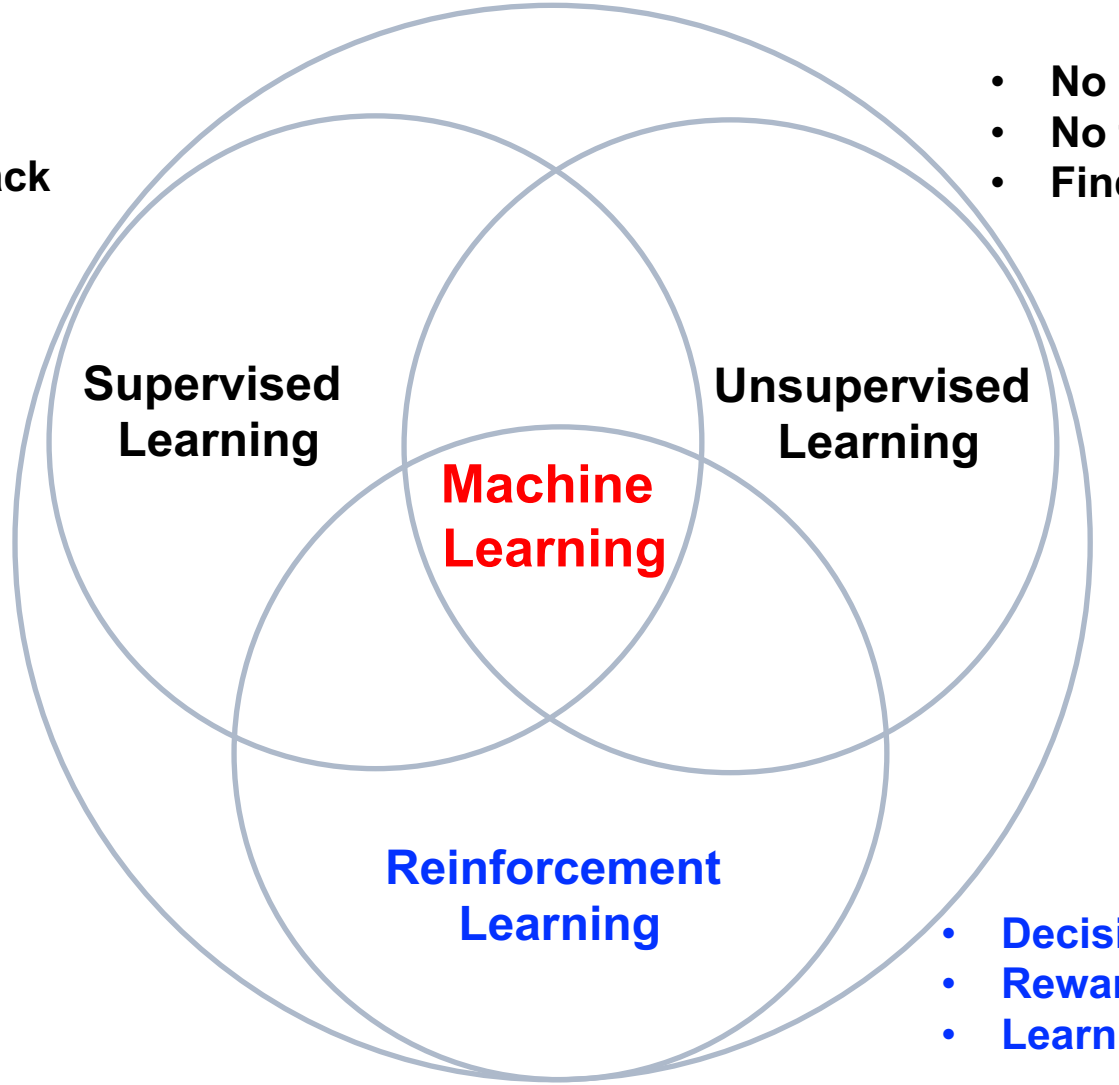


Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Branches of Machine Learning (ML)
Reinforcement Learning (RL)

- Labeled data
- Direct feedback
- Predict

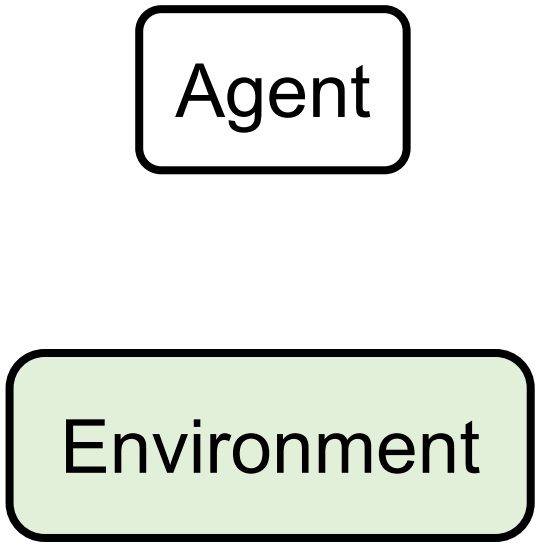
- No Labels
- No feedback
- Find hidden structure



- Decision process
- Reward system
- Learn series of actions

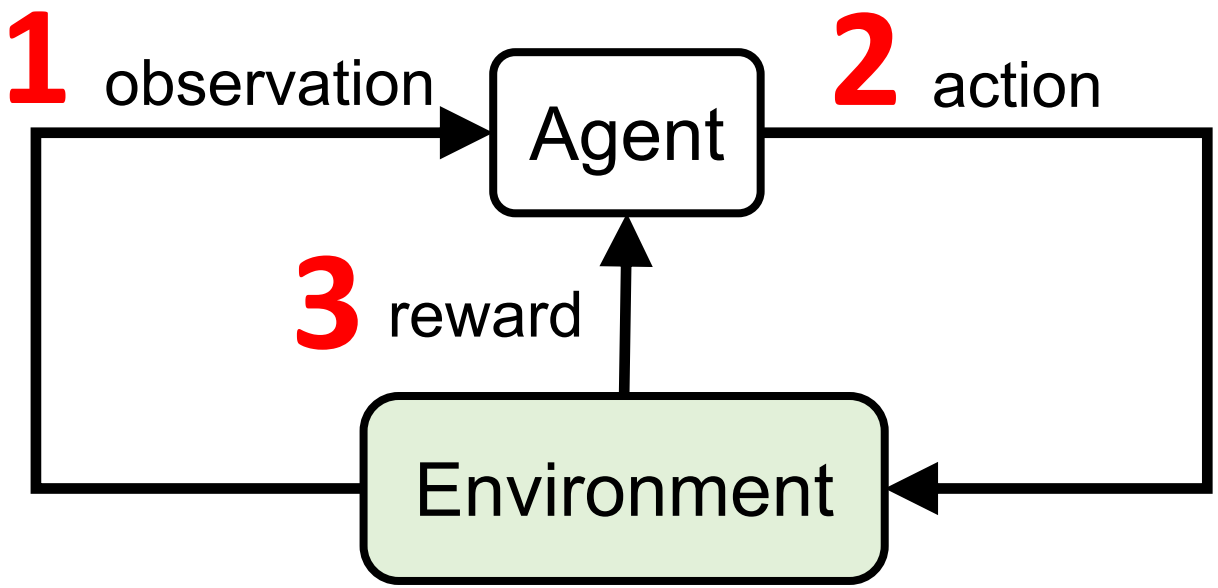
Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning (DL)



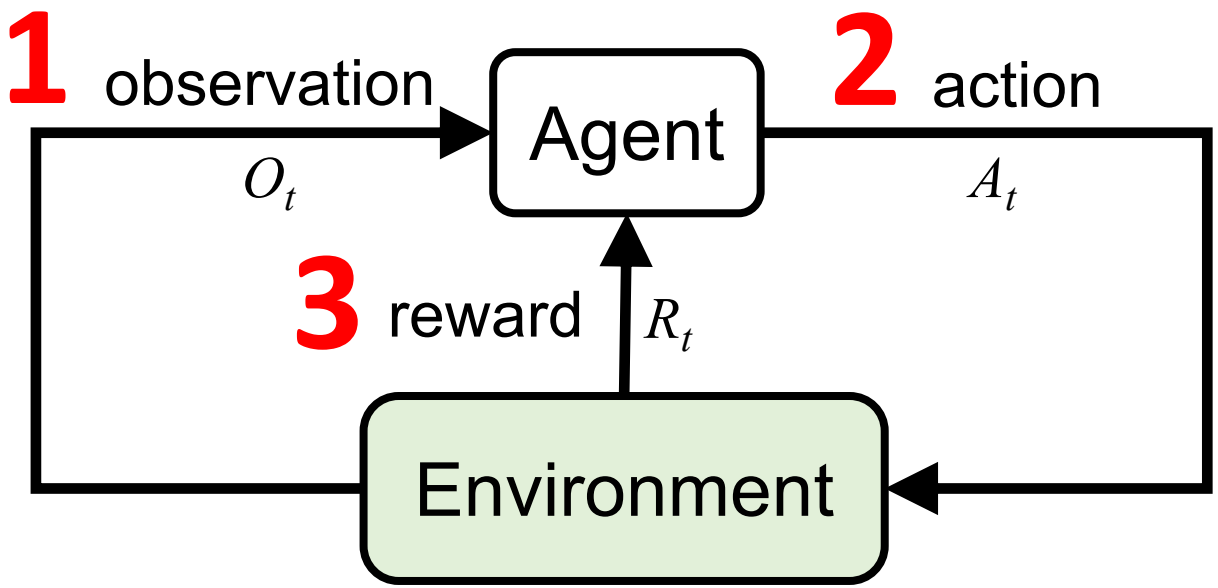
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)



Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

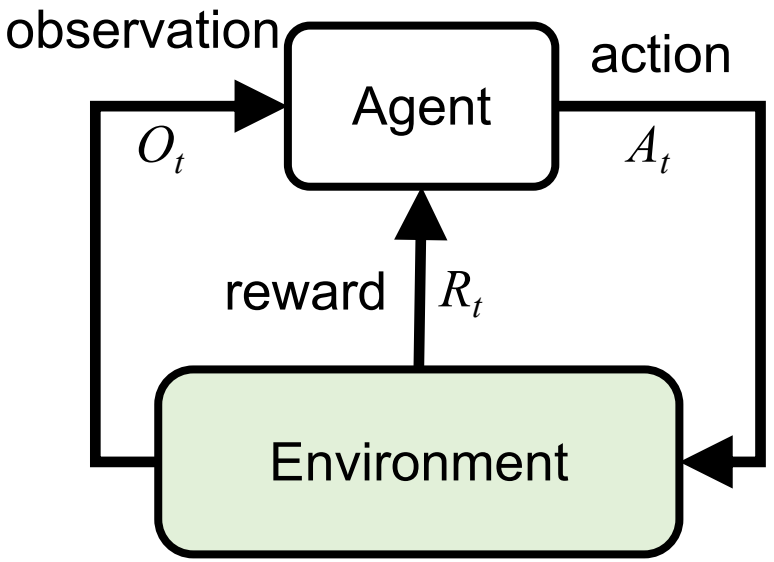
Reinforcement Learning (DL)



Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Agent and Environment

- At each step *t* the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step



Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

David Silver (2015), Introduction to reinforcement learning

- Elementary Reinforcement Learning
  - 1: Introduction to Reinforcement Learning
  - 2: Markov Decision Processes
  - 3: Planning by Dynamic Programming
  - 4: Model-Free Prediction
  - 5: Model-Free Control
- Reinforcement Learning in Practice
  - 6: Value Function Approximation
  - 7: Policy Gradient Methods
  - 8: Integrating Learning and Planning
  - 9: Exploration and Exploitation
  - 10: Case Study: RL in Classic Games

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement learning

- **Reinforcement learning is**
  learning what to do
  —how to map situations to actions
  —so as to maximize a numerical reward signal.

Two most important distinguishing features of reinforcement learning

- **trial-and-error search**
- **delayed reward**

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

History and State

- The history is the sequence of observations, actions, rewards

$$H_t = O_1, A_1, R_1, \ldots, A_{t-1}, O_t, R_t$$

- i.e. all observable variables up to time t
- i.e. the sensorimotor stream of a robot or embodied agent
- What happens next depends on the history:
  - The agent selects actions
  - The environment selects observations/rewards
- State is the information used to determine what happens next
- Formally, state is a function of the history:

$$S_t = f(H_t)$$

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Information State

- **An information state (a.k.a. Markov state) contains all useful information from the history.**
- **Definition**

  A state S_t is Markov if and only if

  $$P[S_{t+1} \mid S_t] = P[S_{t+1} \mid S_1, \ldots, S_t]$$

- **"The future is independent of the past given the present"**

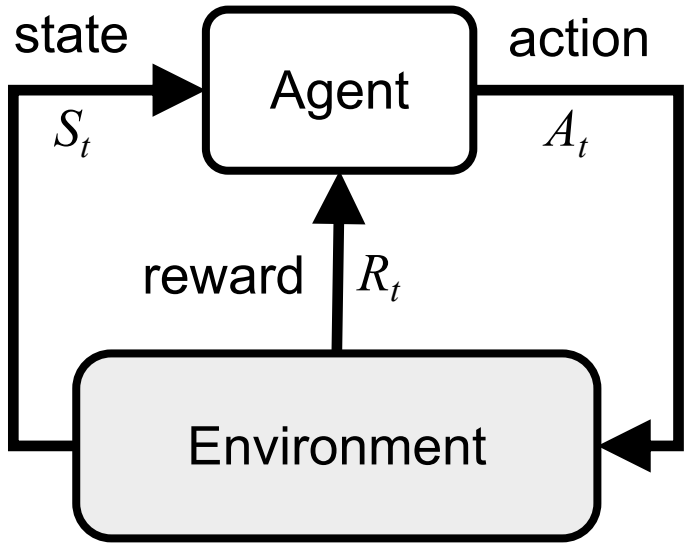
  $$H_{1:t} \rightarrow S_t \rightarrow H_{t+1:\infty}$$

- **Once the state is known, the history may be thrown away i.e. The state is a sufficient statistic of the future**
- **The environment state S_t^e is Markov**
- **The history H_t is Markov**

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Fully Observable Environments

- **Full observability:**
  - **agent directly observes environment state**
  - **Agent state = environment state = information state**
  - **Formally, this is a Markov decision process (MDP)**



Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

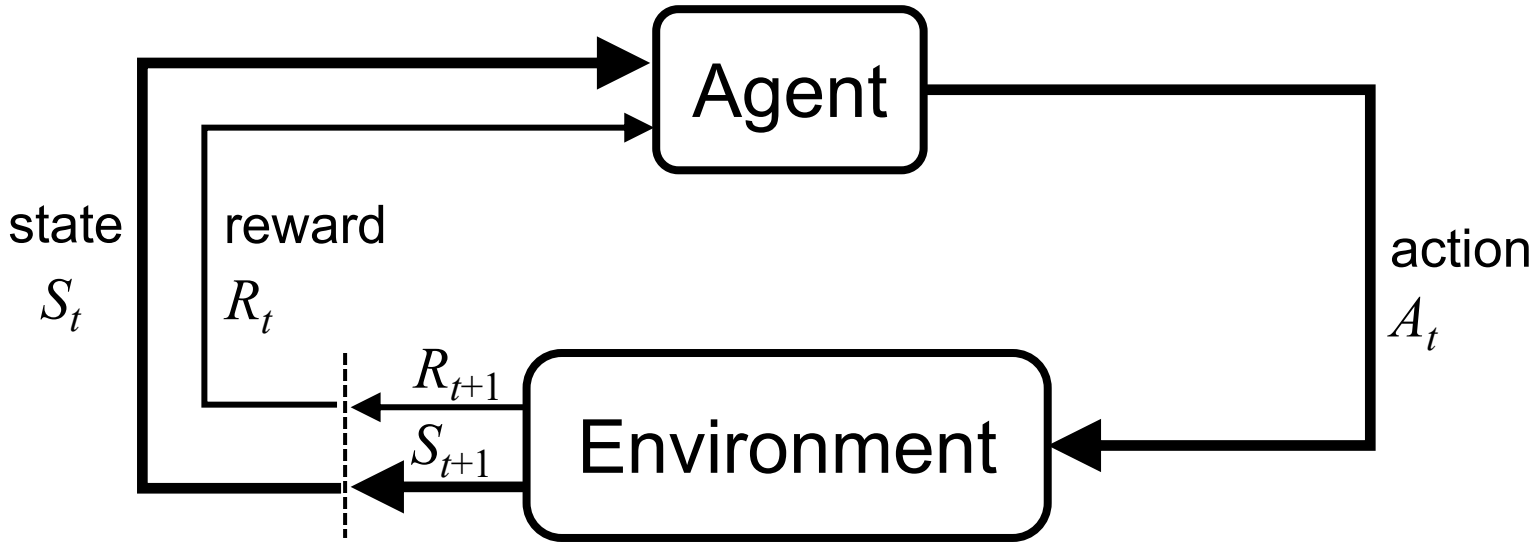
Partially Observable Environments

- **Partial observability**: agent **indirectly** observes environment
  - A robot with camera vision isn't told its absolute location
  - A trading agent only observes current prices
  - A poker playing agent only observes public cards
- Now agent state ≠ environment state
- Formally this is a **partially observable Markov decision process (POMDP)**
- Agent must construct its own state representation S^a_t, e.g.
  - Complete history: $S^a_t = H_t$
  - **Beliefs** of environment state: $S^a_t = (P[S^e_t = s_1],\ldots,P[S^e_t = s_n])$
  - Recurrent neural network: $S^a_t = \sigma(S^a_{t-1} W_s + O_t W_o)$

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning (DL)

The Agent-Environment Interaction in a Markov Decision Process (MDP)



Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Characteristics of Reinforcement Learning

- No supervisor, only a reward signal
- Feedback is delayed, not instantaneous
- Time really matters (sequential, non i.i.d data)
- Agent's actions affect the subsequent data it receives

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Examples of Reinforcement Learning

- **Make a humanoid robot walk**
- **Play may different Atari games better than humans**
- **Manage an investment portfolio**

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Examples of Rewards

- Make a humanoid robot walk
  - +ve reward for forward motion
  - -ve reward for falling over
- Play may different Atari games better than humans
  - +/-ve reward for increasing/decreasing score
- Manage an investment portfolio
  - +ve reward for each $ in bank

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Sequential Decision Making

- Goal: select actions to maximize total future reward
- Actions may have long term consequence
- Reward may be delayed
- It may be better to sacrifice immediate reward to gain more long-term reward
- Examples:
  - A financial investment (may take months to mature)
  - Blocking opponent moves (might help winning chances many moves from now)

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Elements of Reinforcement Learning

- Agent
- Environment
- Policy
- Reward signal
- Value function
- Model

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Elements of Reinforcement Learning

- Policy
  - Agent's behavior
  - It is a map from state to action
- Reward signal
  - The goal of a reinforcement learning problem
- Value function
  - How good is each state and/or action
  - A prediction of future reward
- Model
  - Agent's representation of the environment

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Major Components of an RL Agent

1. **Policy**: agent's behaviour function
2. **Value** function: how good is each state and/or action
3. **Model**: agent's representation of the environment

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Policy

- A policy is the agent's behaviour
- It is a map from state to action, e.g.
  - Deterministic policy: $a = \pi(s)$
  - Stochastic policy: $\pi(a|s) = P[A_t = a|S_t = s]$

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Value Function

- **Value function is a prediction of future reward**
- **Used to evaluate the goodness/badness of states**
- **And therefore to select between actions, e.g.**

$$v_\pi(s) = E_\pi\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots \mid S_t = s\right]$$

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Model

- A **model** predicts what the environment will do next
- P predicts the next state
- R predicts the next (immediate) reward, e.g.

$$P^{a}_{ss'} = P[S_{t+1} = s' \mid S_{t+1} = s, A_t = a]$$

$$R^{a}_{s} = E[R_{t+1} \mid S_t = s, A_t = a]$$

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning

- **Value Based**
  - **No Policy (Implicit)**
  - **Value Function**
- **Policy Based**
  - **Policy**
  - **No Value Function**
- **Actor Critic**
  - **Policy**
  - **Value Function**

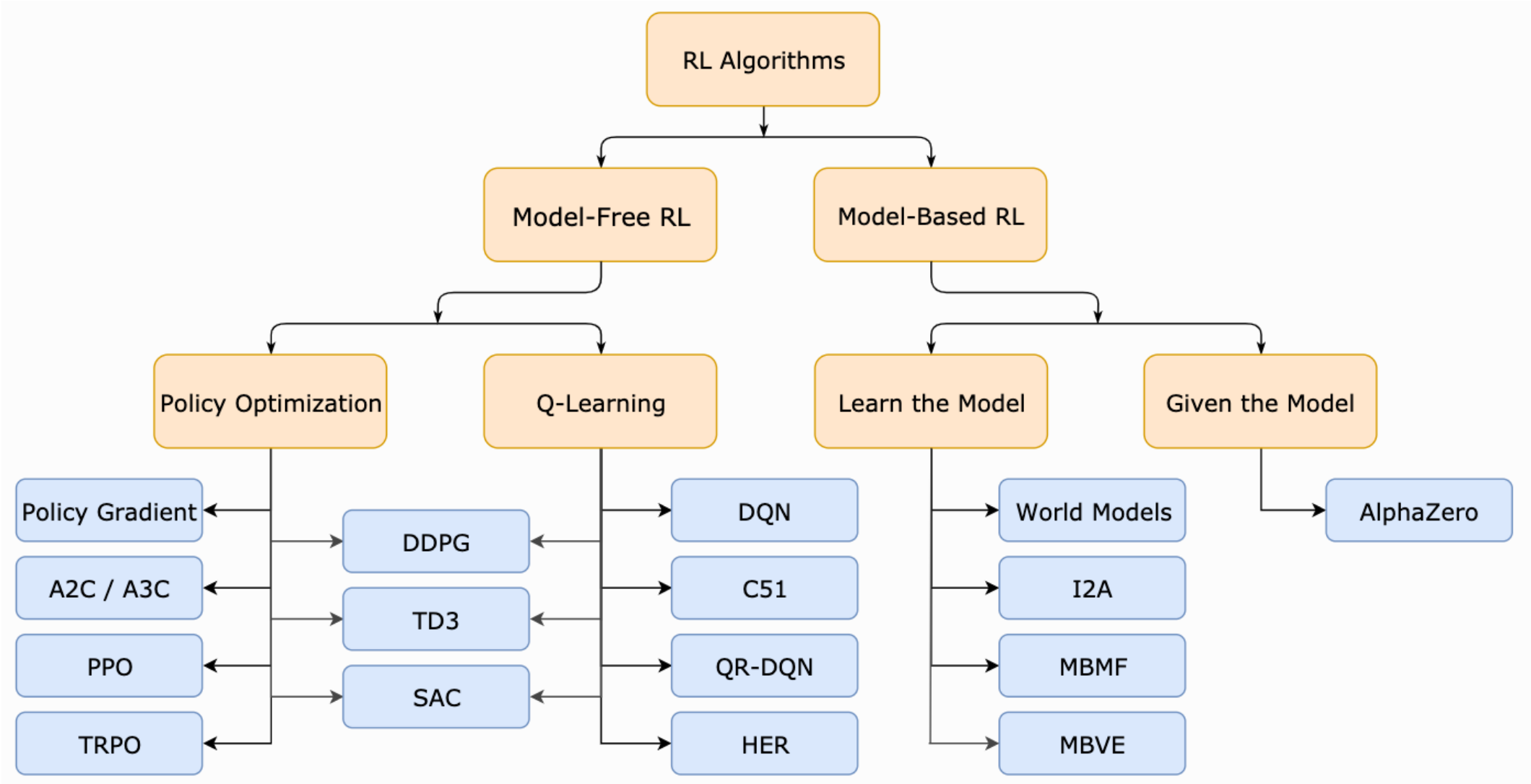
Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning

- Model Free
  - Policy and/or Value Function
  - No Model
- Model Based
  - Policy and/or Value Function
  - Model

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning (RL)
A Taxonomy of RL Algorithms



Source: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html

Learning and Planning

- **Two fundamental problems in sequential decision making**
  - **Reinforcement Learning**
    - The environment is initially unknown
    - The agent interacts with environment
    - The agent improves its policy
  - **Planning**
    - A model of the environment is known
    - The agent performs computations with its model (without any external interaction)
    - The agent improves its policy
    - a.k.a deliberation, reasoning, introspection, pondering, thought, search

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Exploration and Exploitation

- Reinforcement learning is like trial-and-error learning
- The agent should discover a good policy
- From its experiences of the environment
- Without losing too much reward along the way
- Exploration finds more information about the environment
- Exploitation exploits known information to maximise reward
- It is usually important to explore as well as exploit

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Exploration and Exploitation Examples

- Restaurant Selection
  - Exploitation: Go to your favorite restaurant
  - Exploration: Try a new restaurant
- Online Banner Advertisements
  - Exploitation: Show the most successful advert
  - Exploration: Show a different advert

Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Exploration and Exploitation Examples

- Oil Drilling
  - Exploitation: Drill at the best known location
  - Exploration: Drill at a new location
- Game Playing
  - Exploitation: Play the move you believe is best
  - Exploration: Play an experimental move

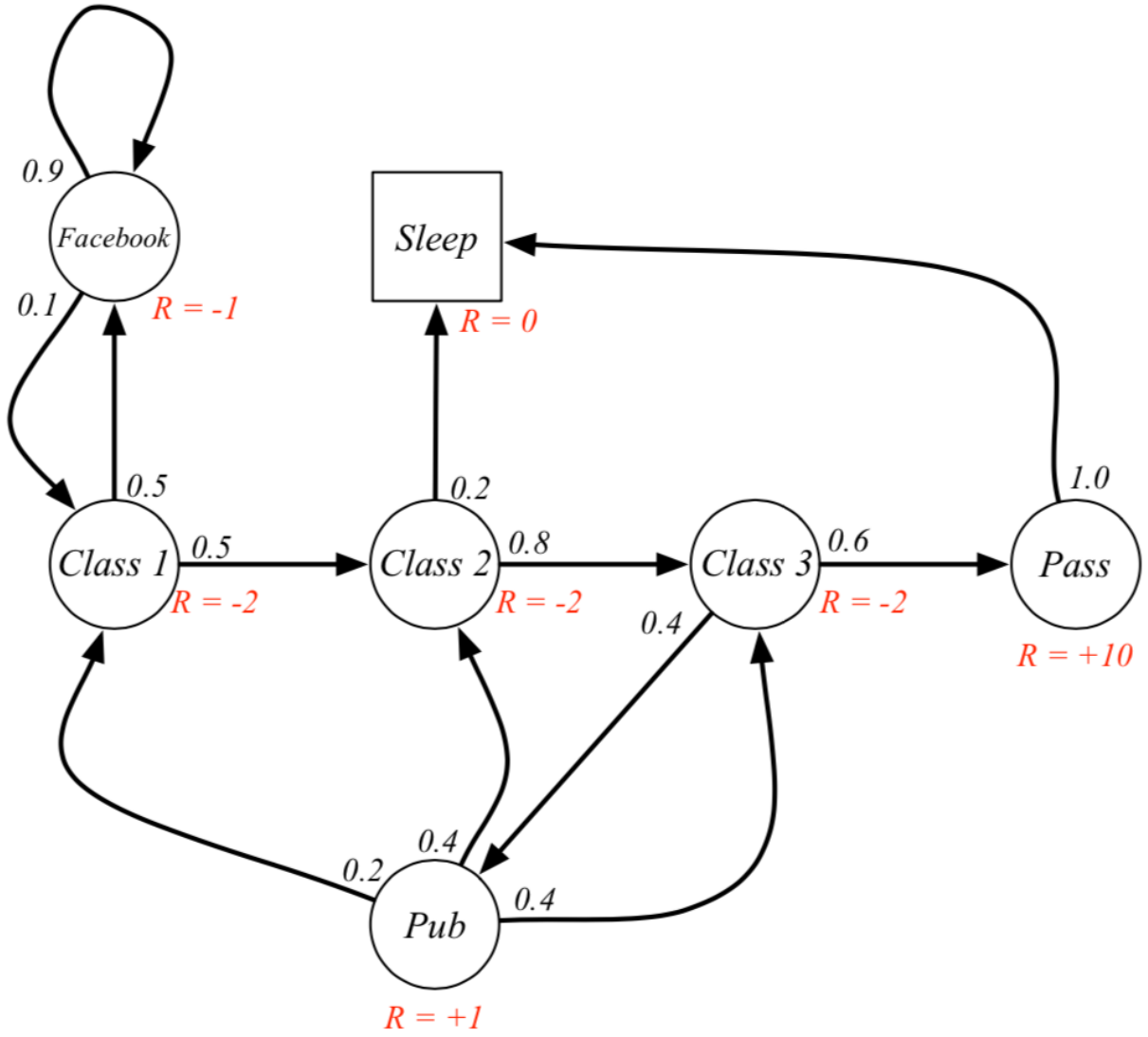
Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Prediction and Control

- **Prediction: evaluate the future**
  - **Given a policy**
- **Control: optimize the future**
  - **Find the best policy**

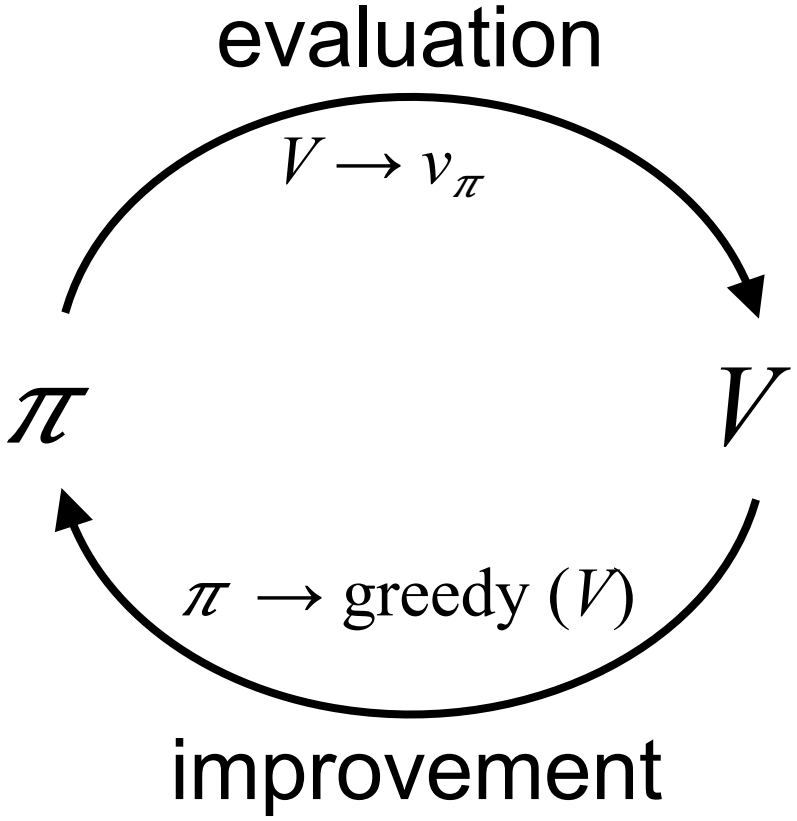
Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

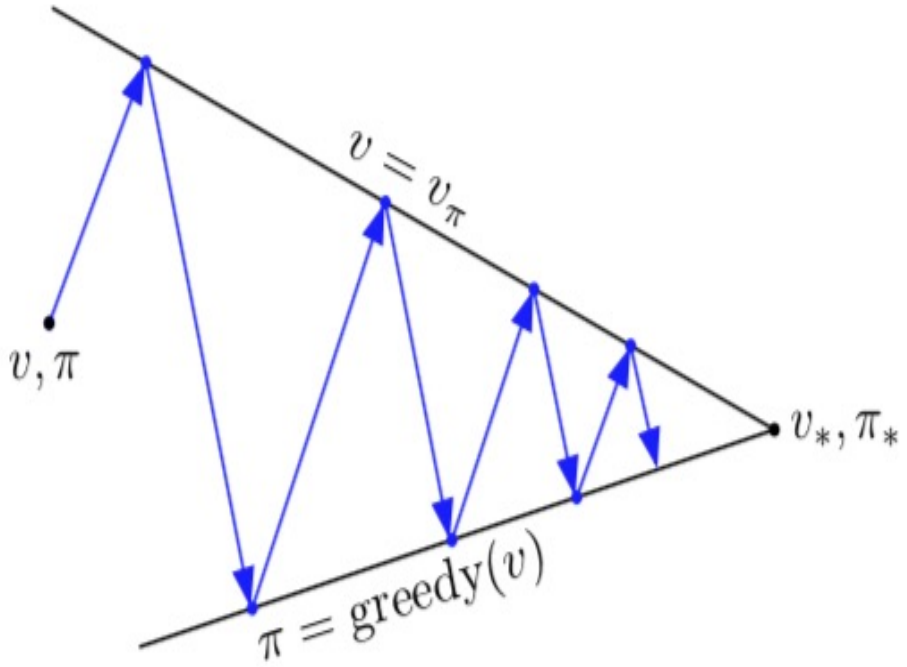
Markov Decision Processes (MDP)
Example: Student MDP



Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Generalized Policy Iteration (GPI)

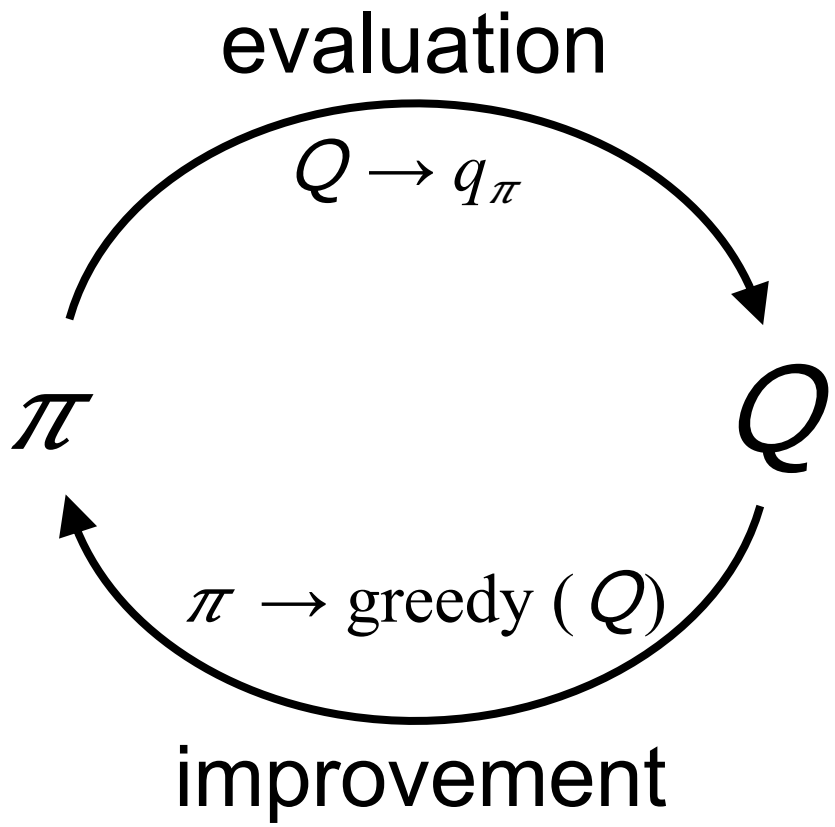




$$\pi_* \rightleftharpoons v_*$$

Generalized Policy Iteration (GPI)

Any iteration of **policy evaluation** and **policy improvement**, independent of their granularity.



Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

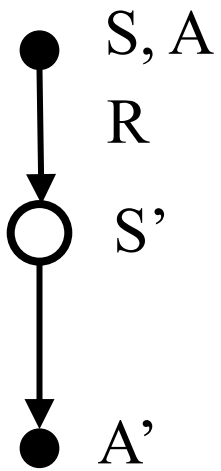
Temporal-Difference (TD) Learning

- **SARSA: On-policy TD Control**
- **Q-learning: Off-policy TD Control**

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

SARSA
(state-action-reward-state-action)
On-policy TD Control

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \; Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)]$$

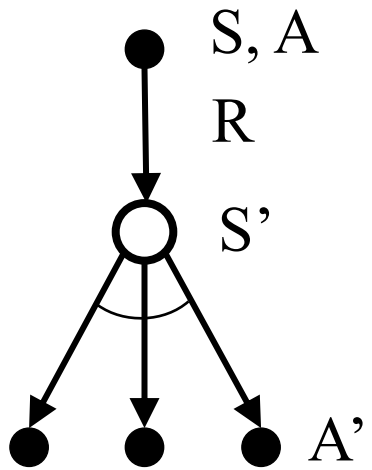


SARSA

Q-learning (Watkins, 1989)

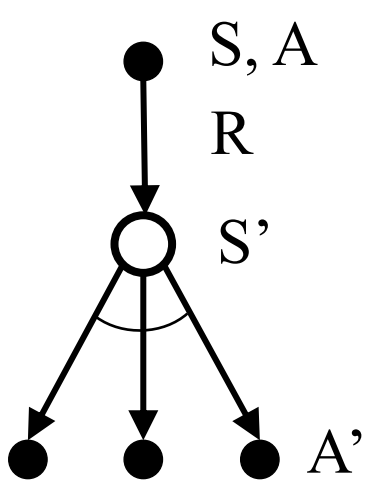
Off-policy TD Control

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max Q(S_{t+1}, a) - Q(S_t, A_t)]$$

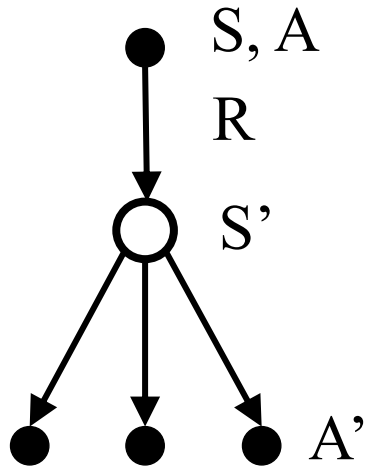


Q-learning

Q-learning and Expected SARSA



Q-learning



Expected SARSA

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Q-learning and Double Q-learning

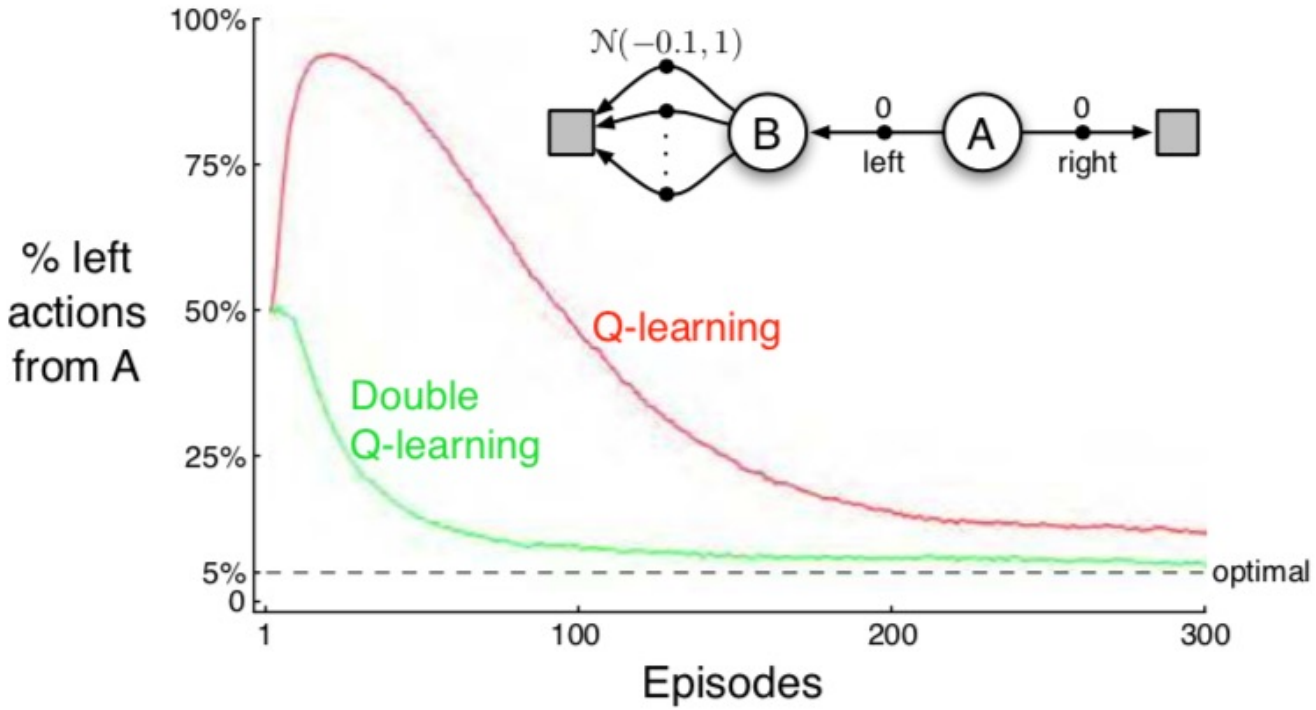


Figure 6.5: Comparison of Q-learning and Double Q-learning on a simple episodic MDP (shown inset). Q-learning initially learns to take the left action much more often than the right action, and always takes it significantly more often than the 5% minimum probability enforced by ε-greedy action selection with $\varepsilon = 0.1$. In contrast, Double Q-learning is essentially unaffected by maximization bias. These data are averaged over 10,000 runs. The initial action-value estimates were zero. Any ties in ε-greedy action selection were broken randomly.

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

n-step methods for sate-action value

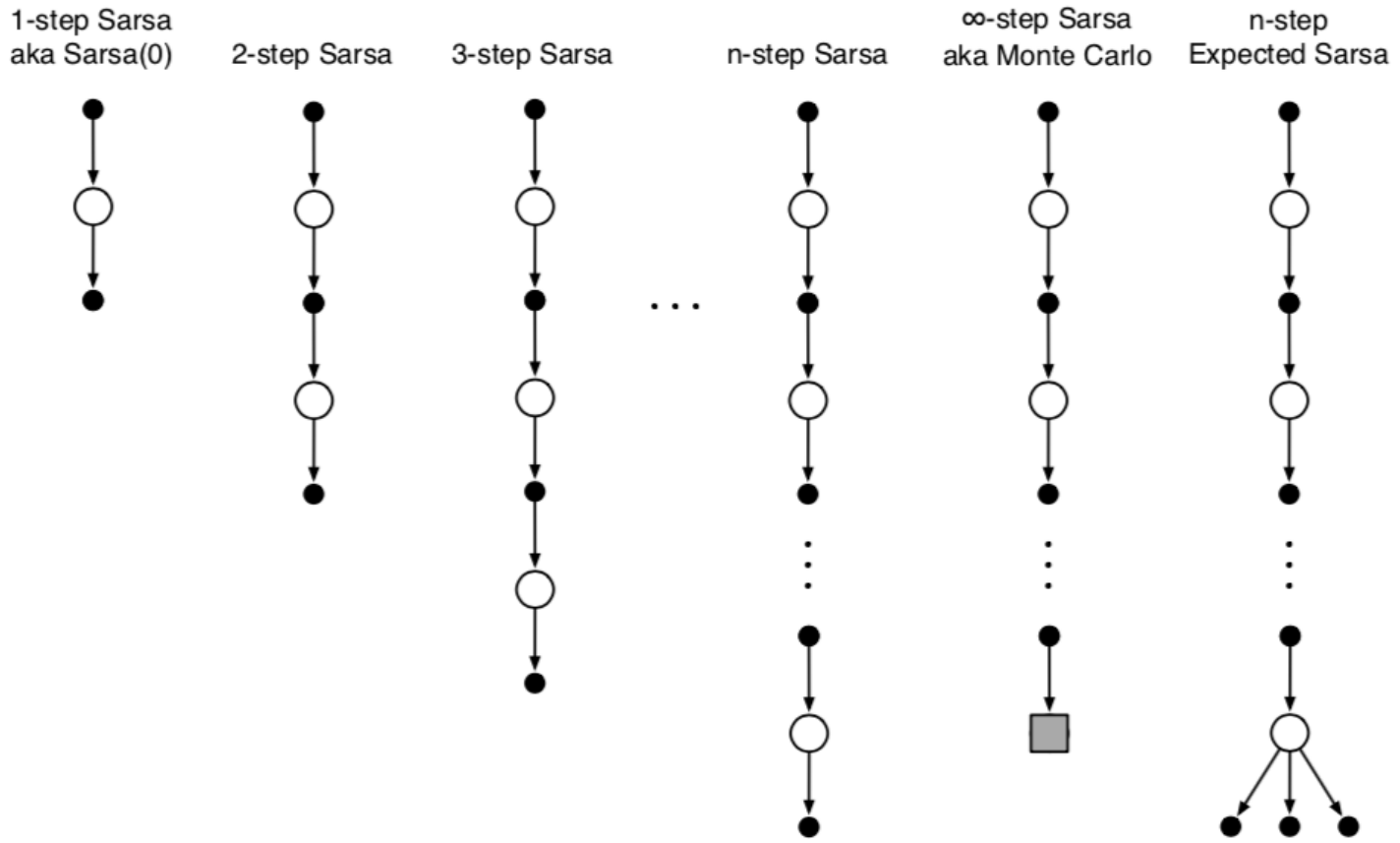
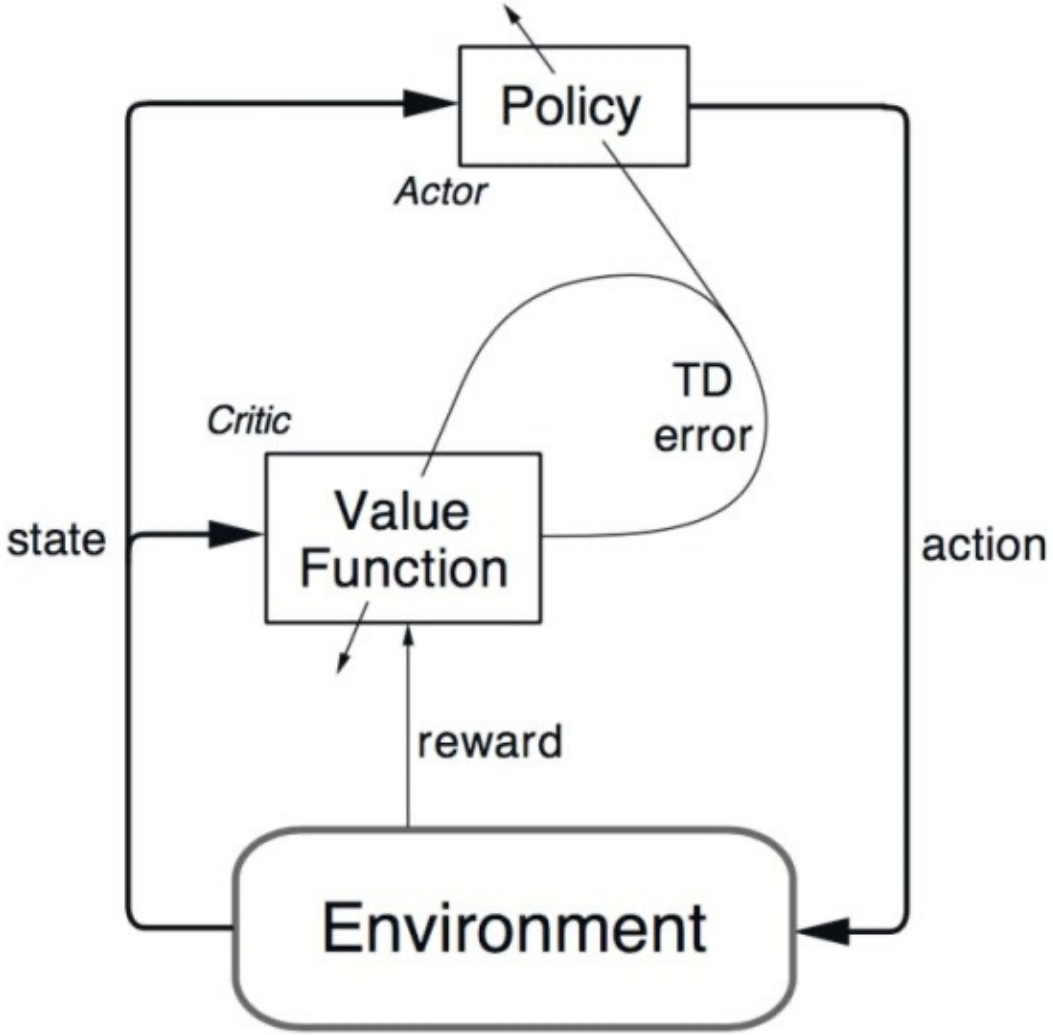


Figure 7.3: The backup diagrams for the spectrum of n-step methods for state–action values. They range from the one-step update of Sarsa(0) to the up-until-termination update of the Monte Carlo method. In between are the n-step updates, based on n steps of real rewards and the estimated value of the nth next state–action pair, all appropriately discounted. On the far right is the backup diagram for n-step Expected Sarsa.

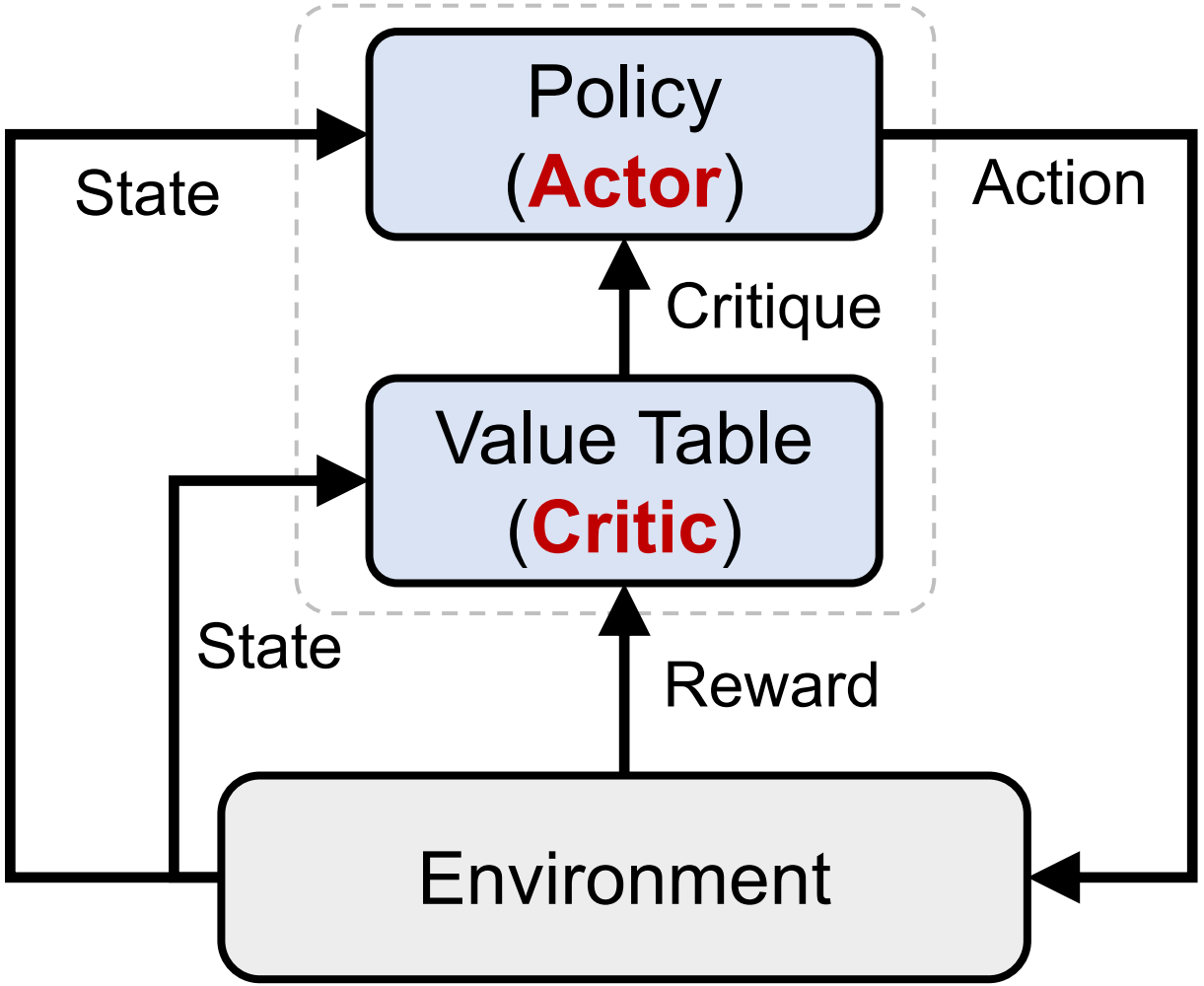
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning
Actor-Critic (AC) Architecture



Source: https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html

Reinforcement Learning
Actor-Critic (AC) Learning Methods



Reinforcement Learning Methods

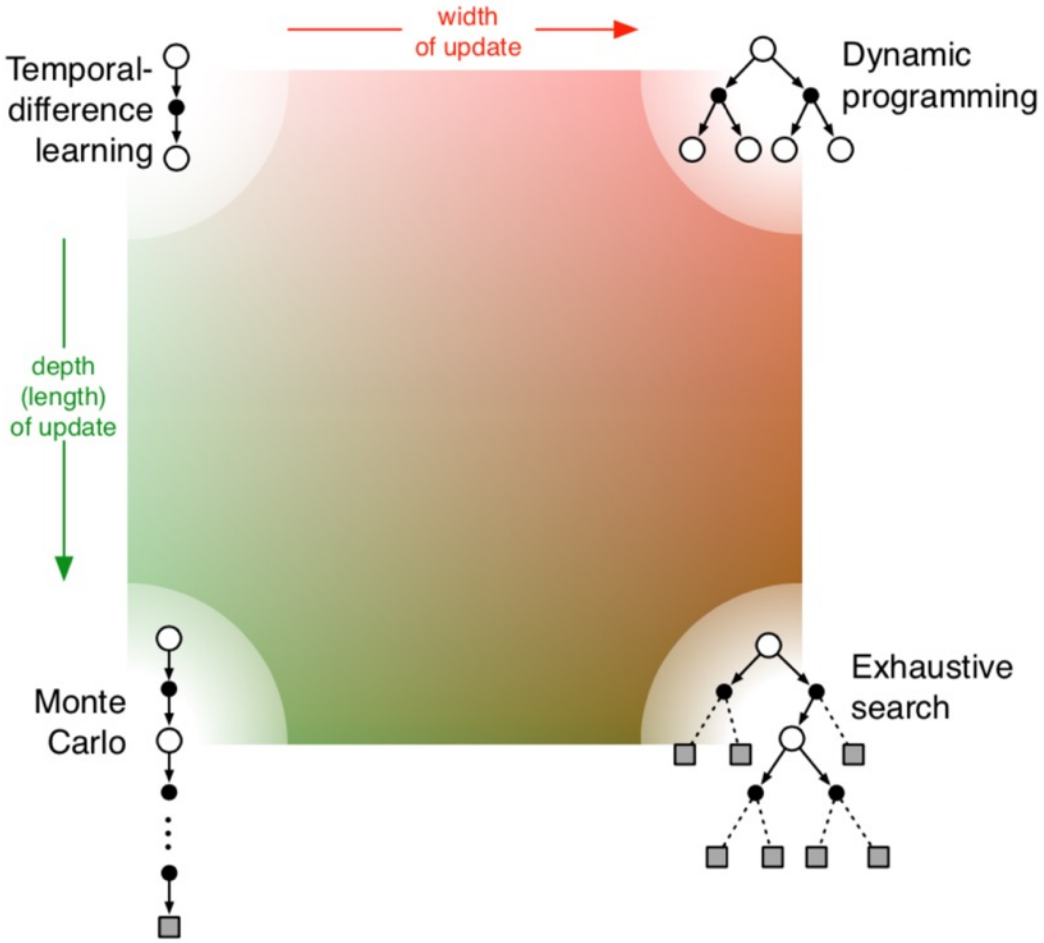


Figure 8.11: A slice through the space of reinforcement learning methods, highlighting the two of the most important dimensions explored in Part I of this book: the depth and width of the updates.

Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Monte Carlo Tree Search (MCTS)

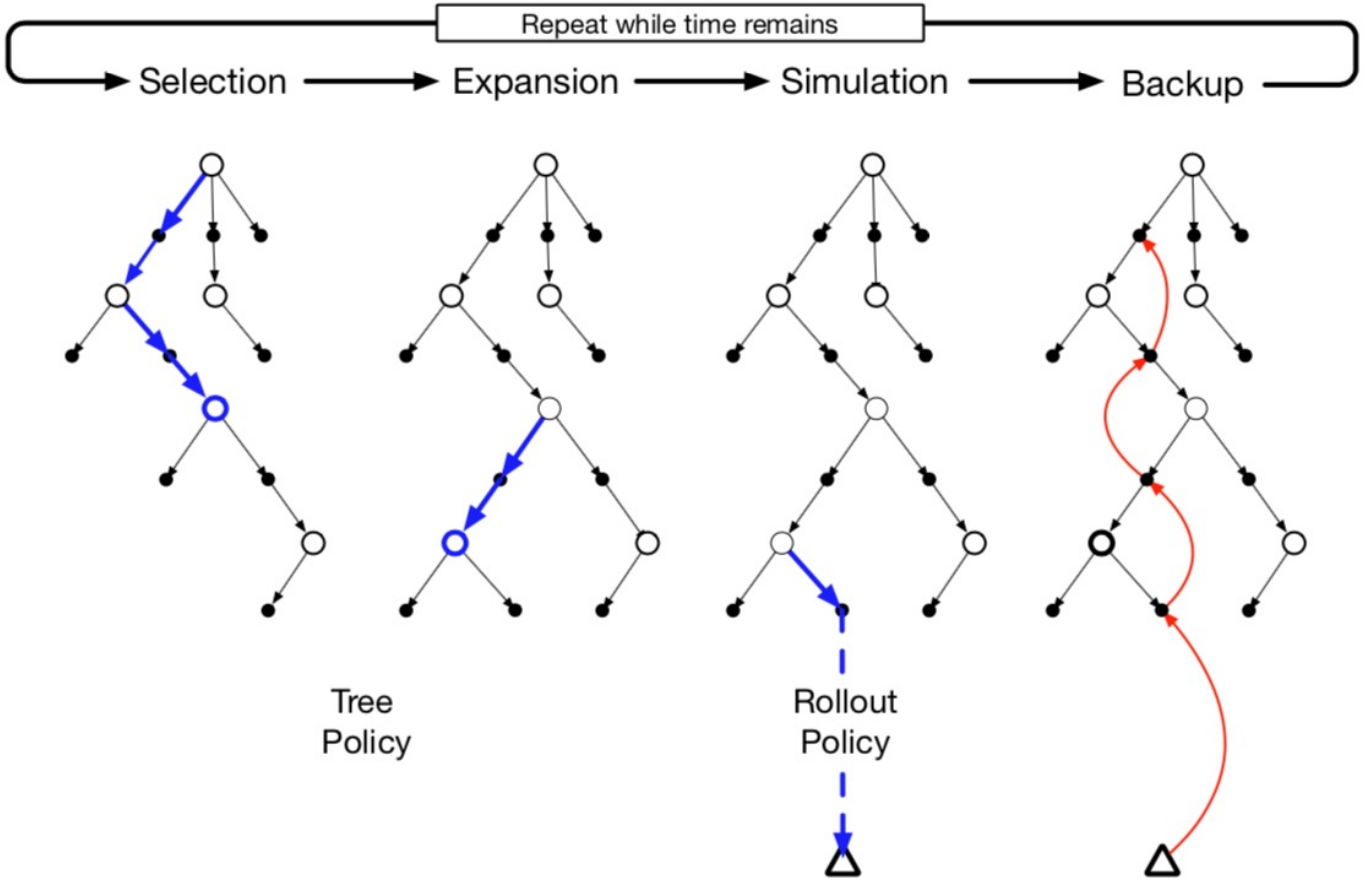
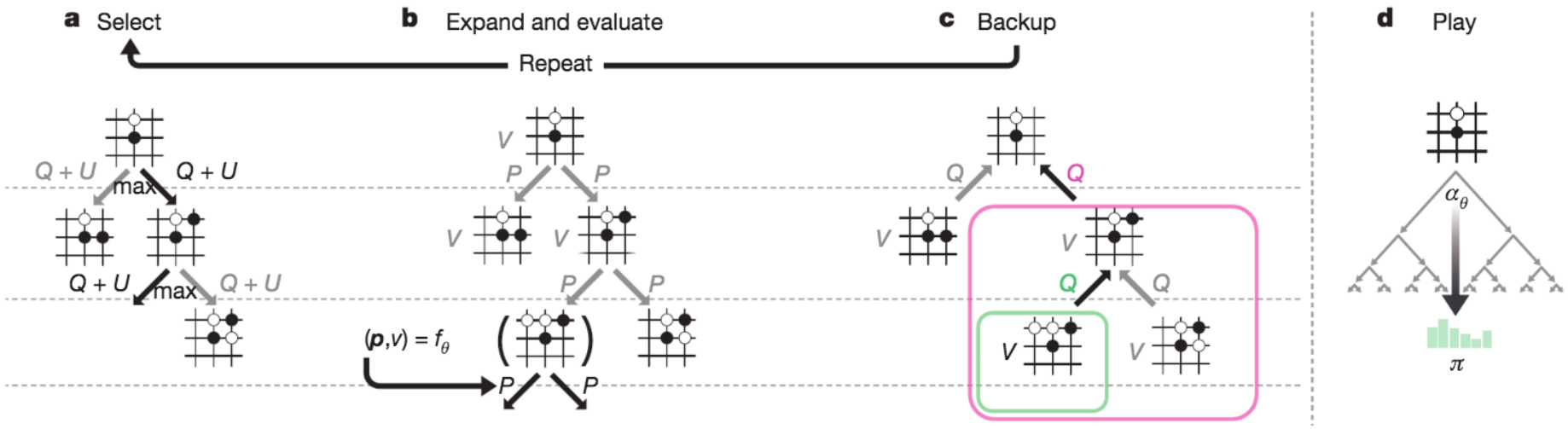


Figure 8.10: Monte Carlo Tree Search. When the environment changes to a new state, MCTS executes as many iterations as possible before an action needs to be selected, incrementally building a tree whose root node represents the current state. Each iteration consists of the four operations **Selection**, **Expansion** (though possibly skipped on some iterations), **Simulation**, and **Backup**, as explained in the text and illustrated by the bold arrows in the trees. Adapted from Chaslot, Bakkes, Szita, and Spronck (2008).

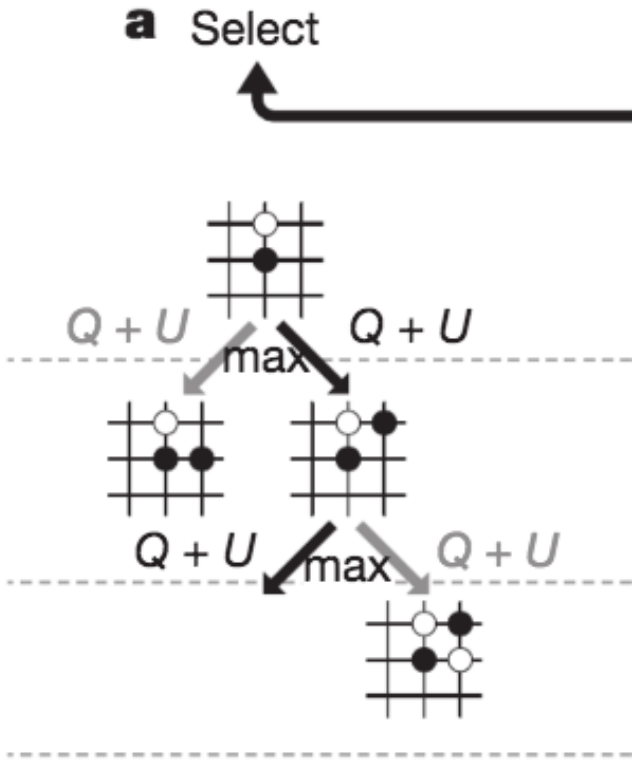
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Monte Carlo Tree Search (MCTS)
MCTS in AlphaGo Zero



Source: David Silver et al. (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.

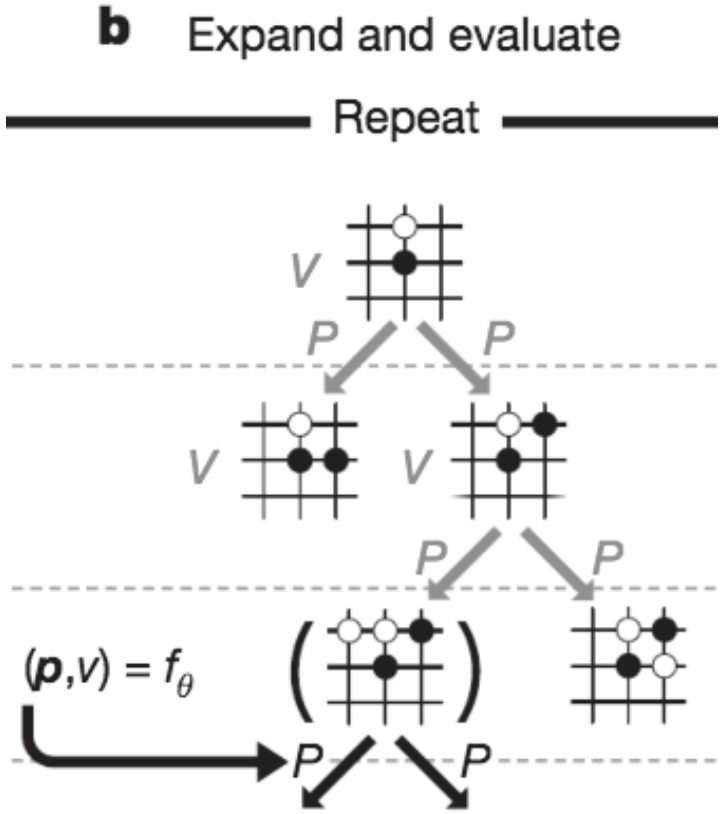
MCTS in AlphaGo Zero



a: Each simulation traverses the tree by selecting the edge with maximum action value Q, plus an upper confidence bound U that depends on a stored prior probability P and visit count N for that edge (which is incremented once traversed).

Source: David Silver et al. (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.

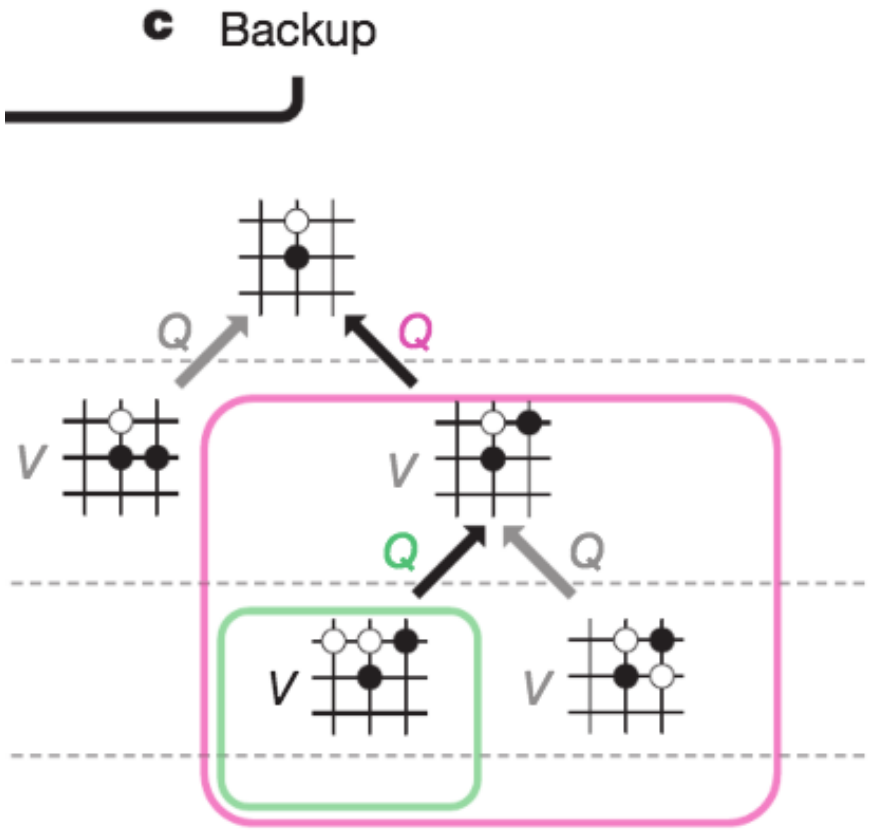
MCTS in AlphaGo Zero



b: The leaf node is expanded and the associated position s is evaluated by the neural network $(P(s, \cdot), V(s)) = f_\theta(s)$; the vector of P values are stored in the outgoing edges from s.

Source: David Silver et al. (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.

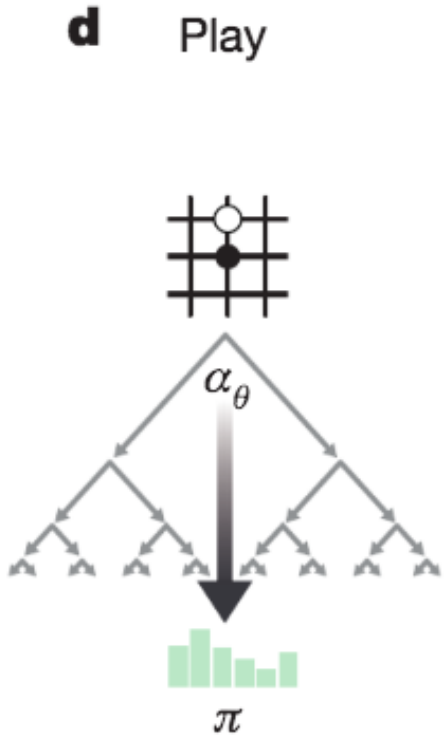
MCTS in AlphaGo Zero



c: Action value Q is updated to track the mean of all evaluations V in the subtree below that action

Source: David Silver et al. (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.

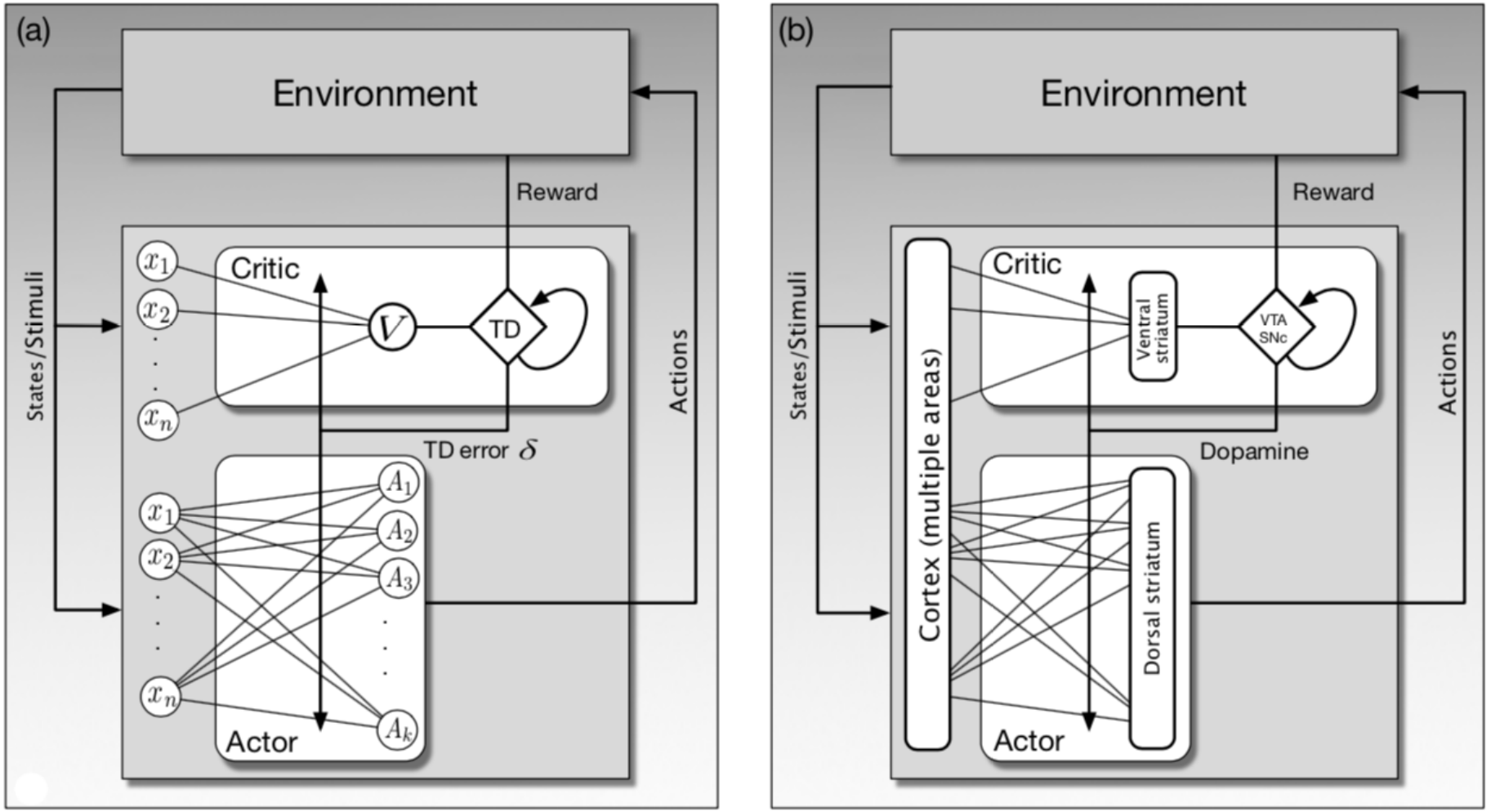
MCTS in AlphaGo Zero



d: Once the search is complete, search probabilities π are returned, proportional to $N^{1/\tau}$, where N is the visit count of each move from the root state and τ is a parameter controlling temperature.

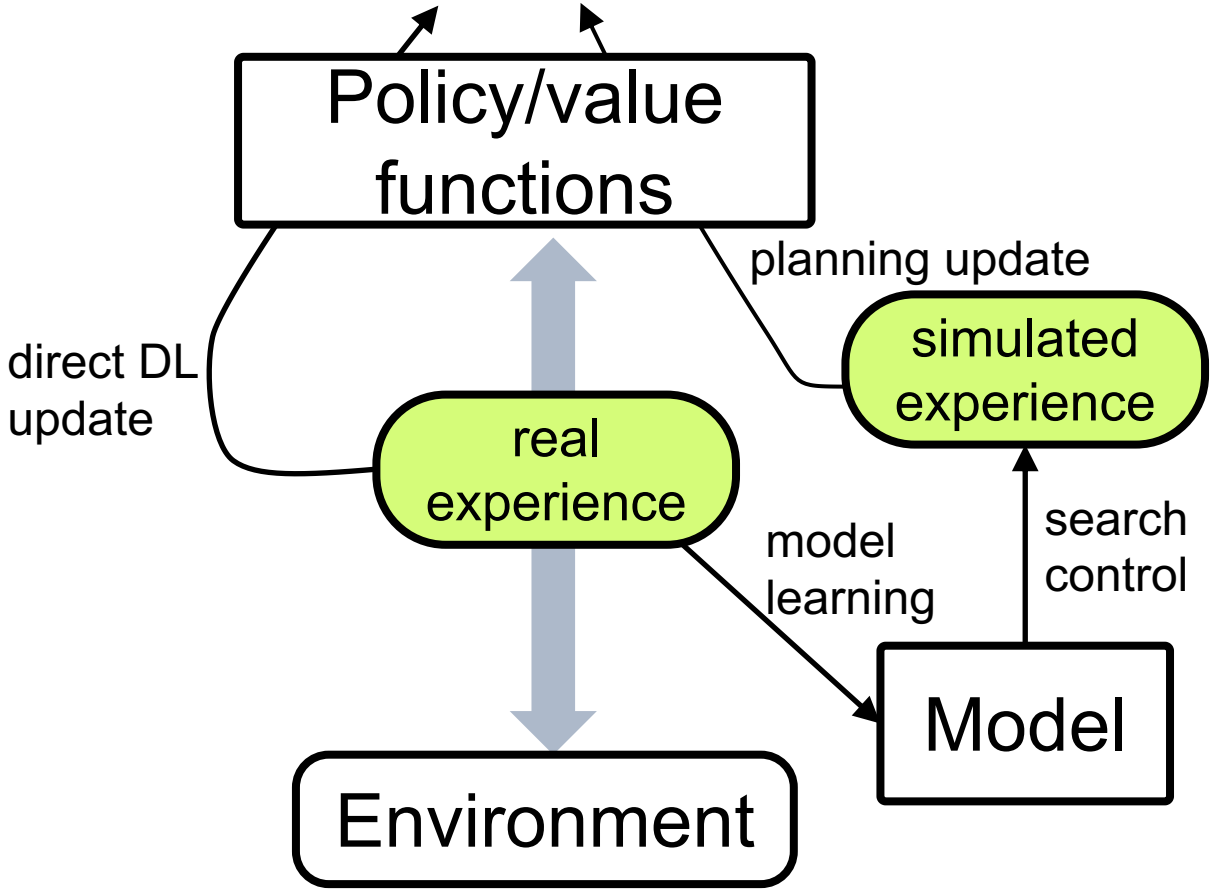
Source: David Silver et al. (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.

Reinforcement Learning
Actor Critic ANN



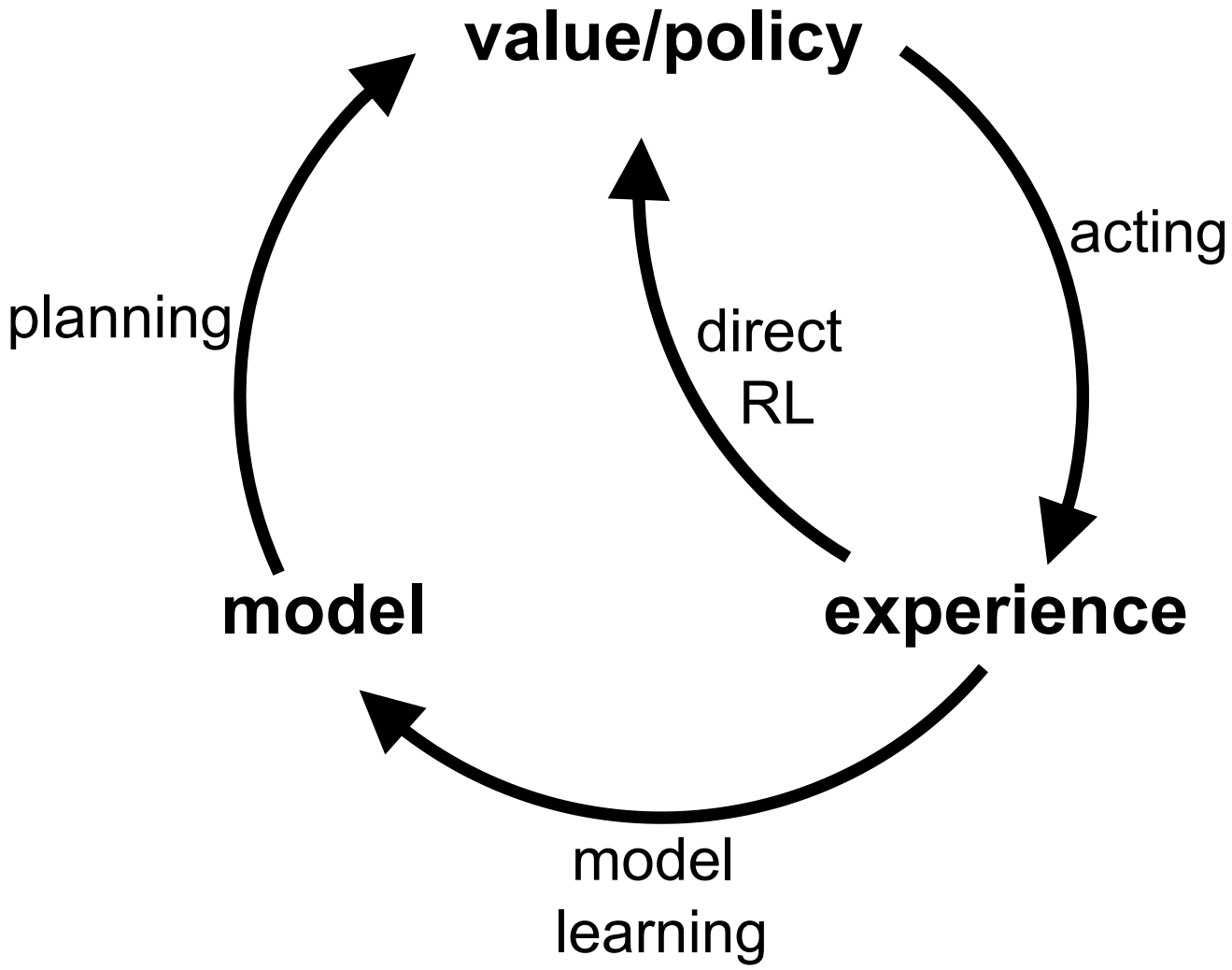
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning
General Dyna Architecture



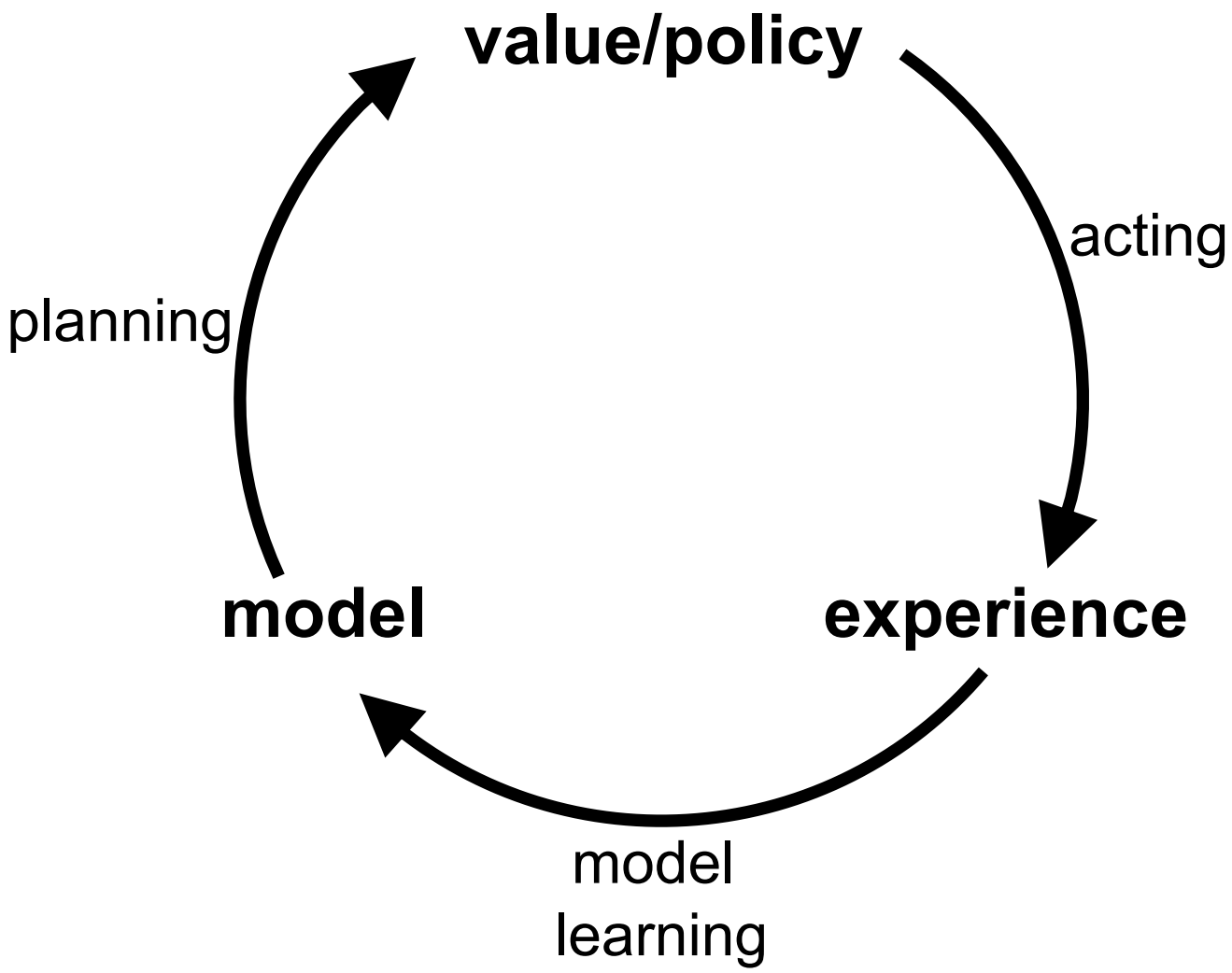
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Dyna:
Integrated Planning, Acting, and Learning



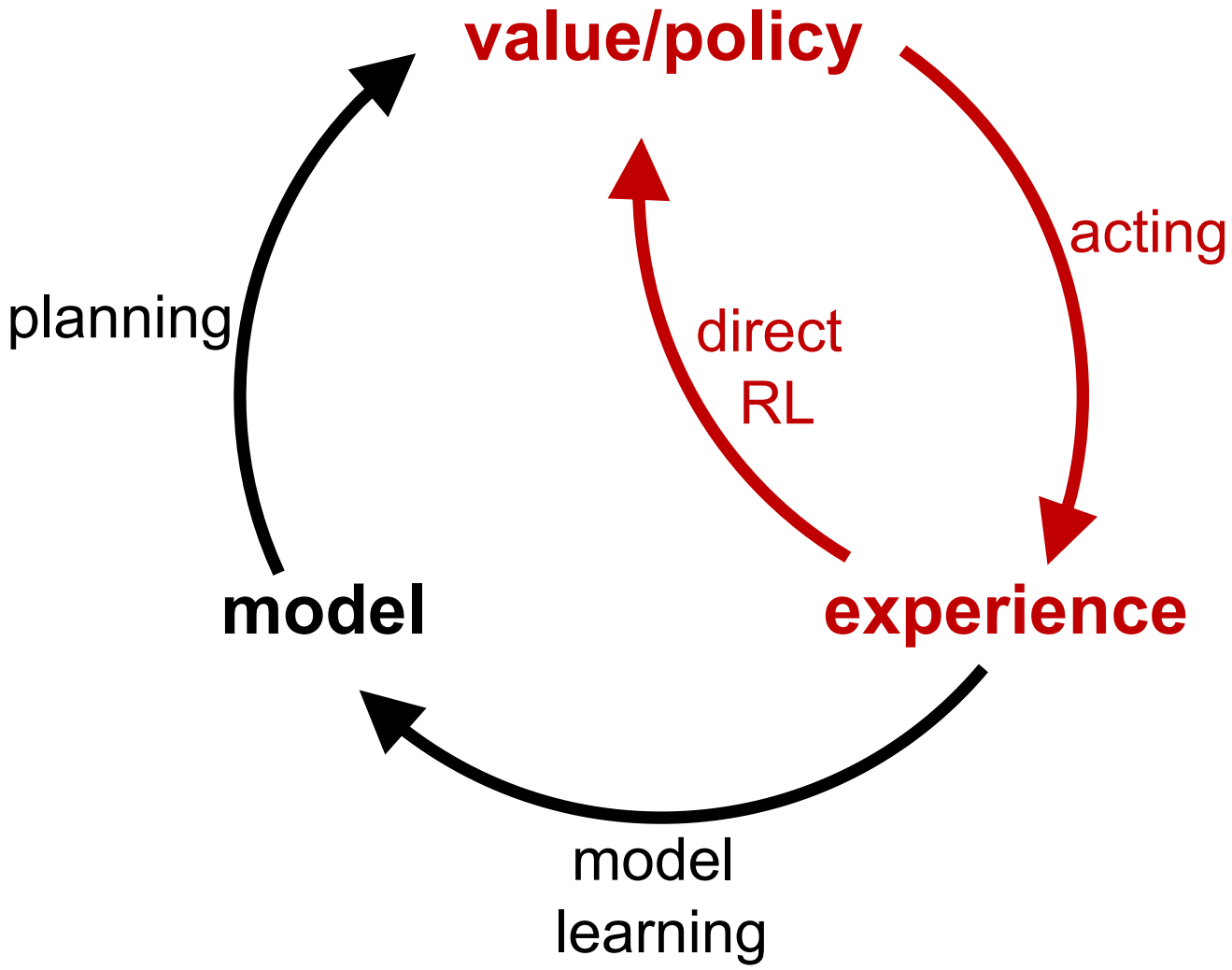
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Model-Based RL



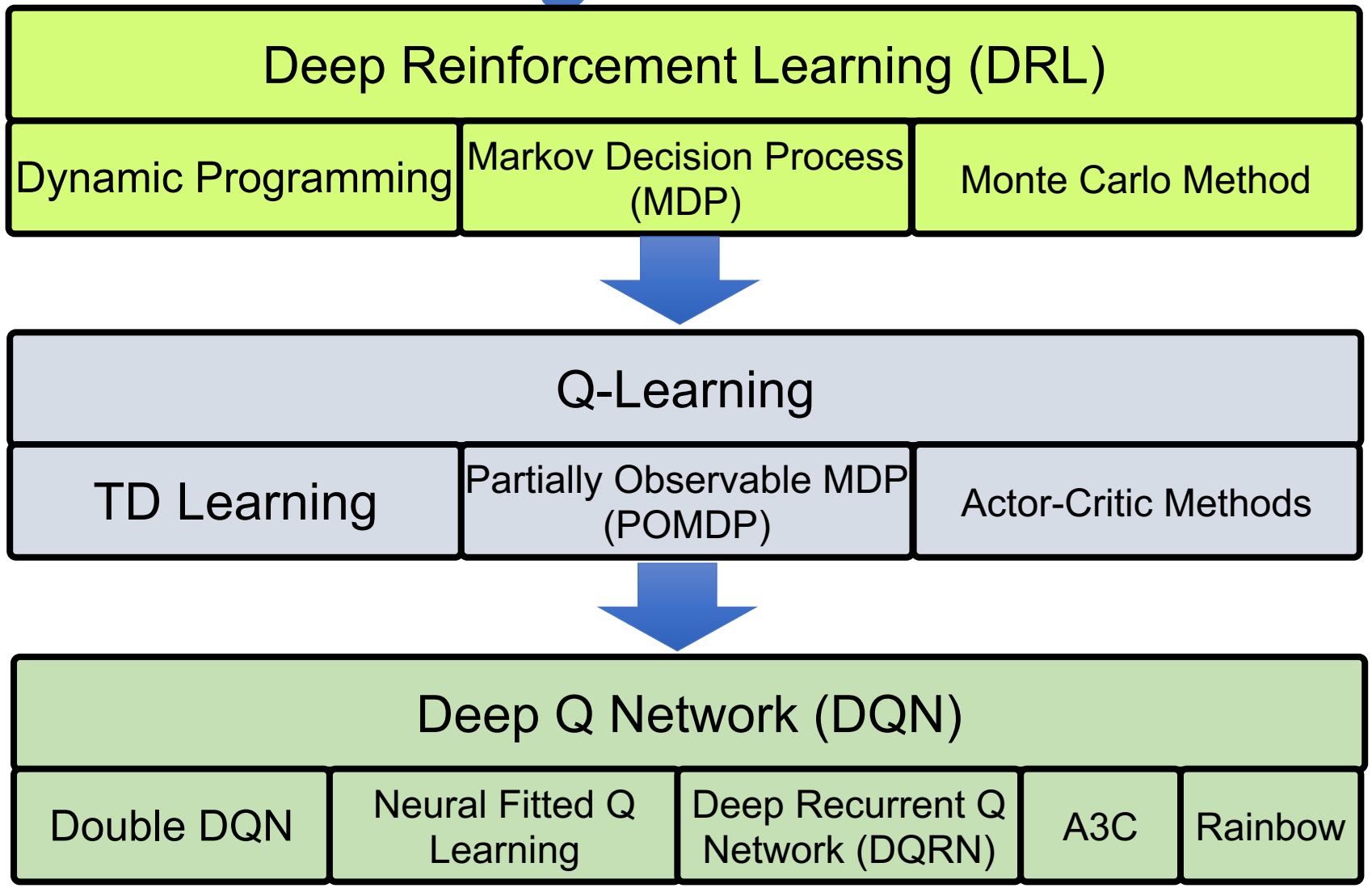
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Model-Free RL (DQN, A3C)



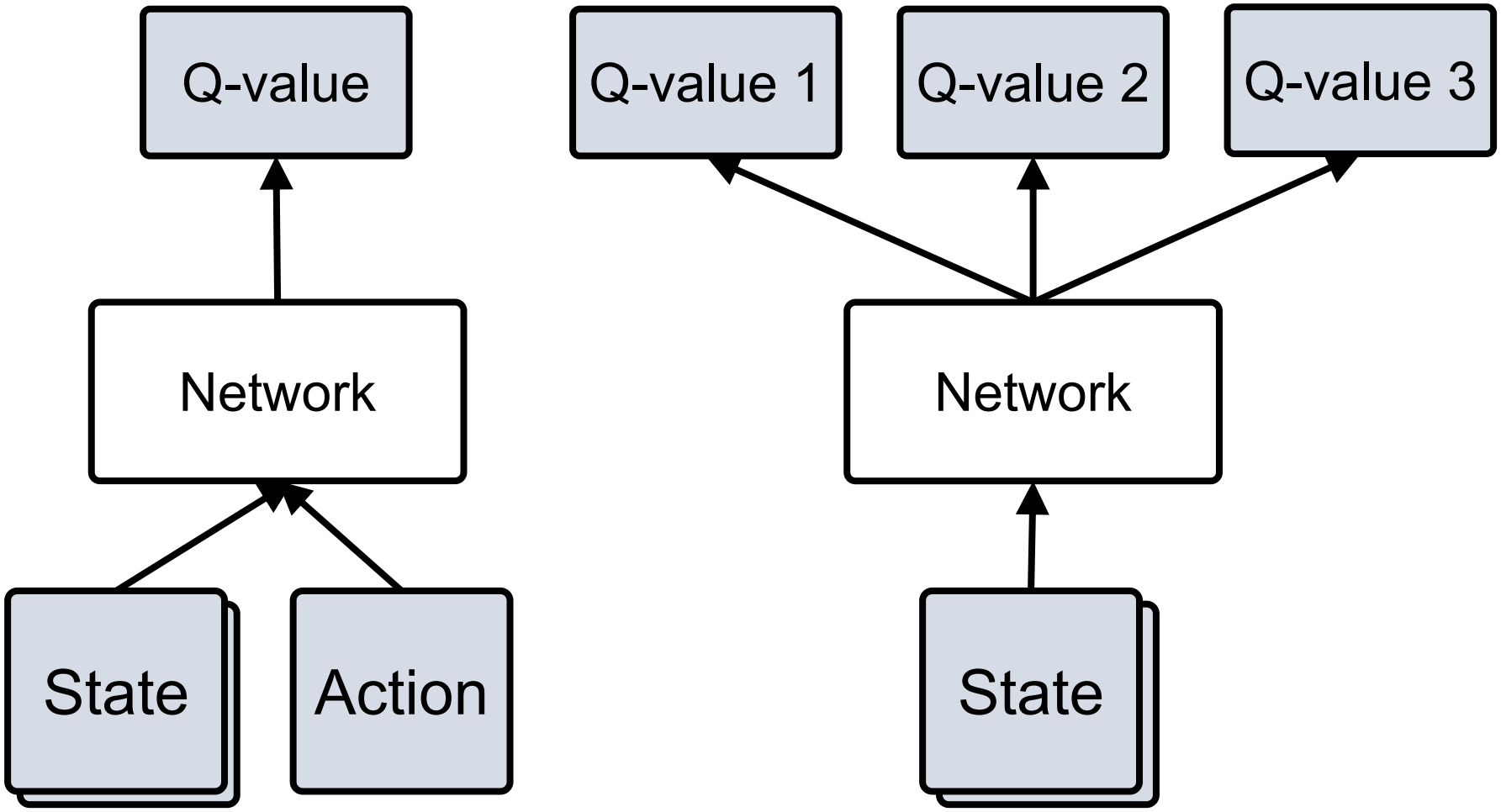
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning Algorithms



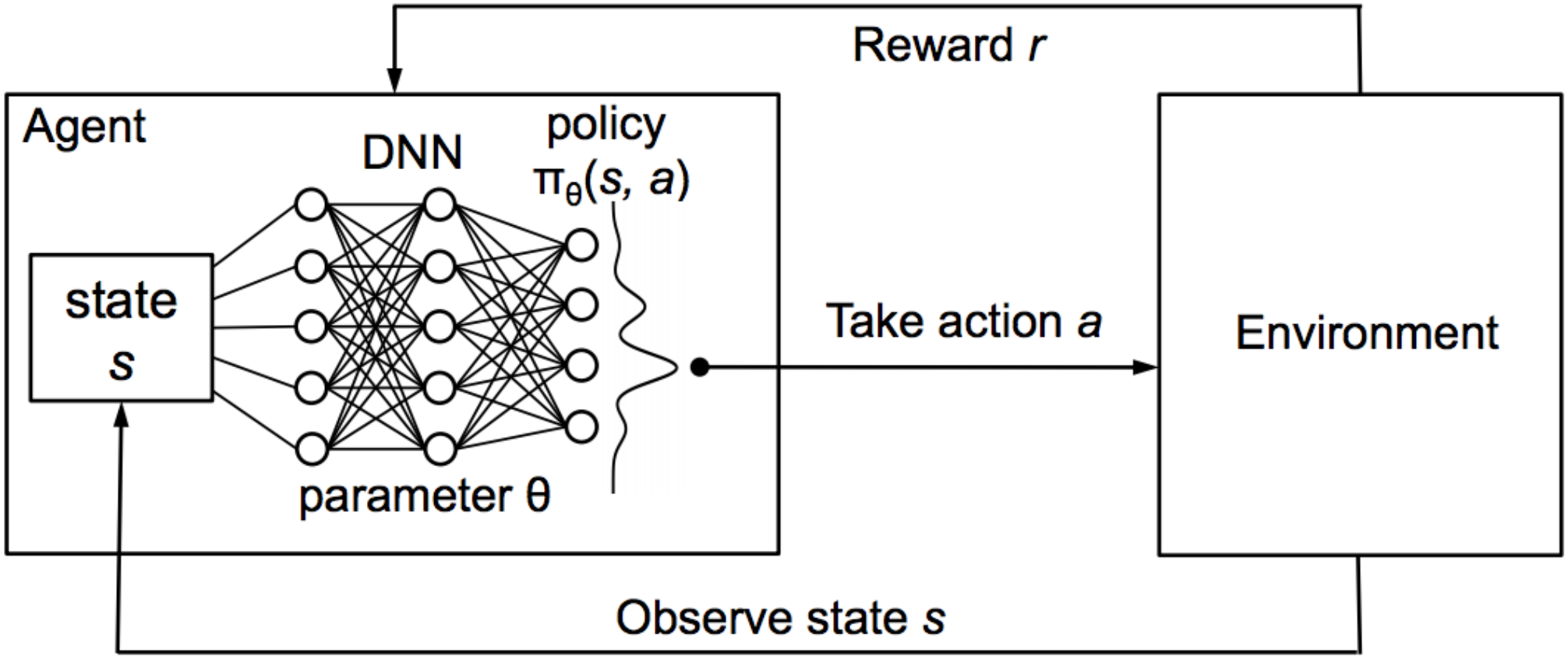
Source: https://amitray.com/deep-learning-past-present-and-future-a-review/

Deep Q-Network (DQN)



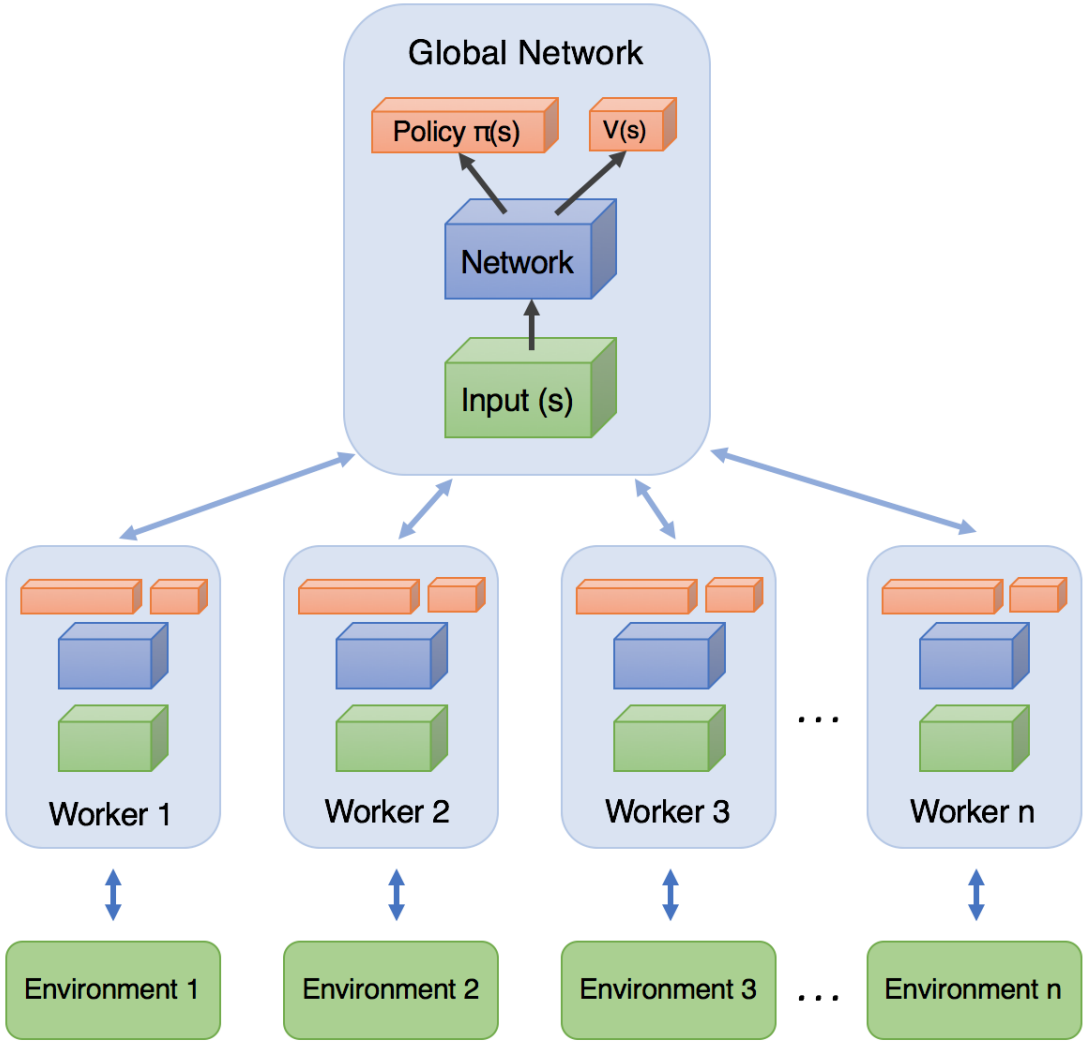
Source: https://neuro.cs.ut.ee/demystifying-deep-reinforcement-learning/

Reinforcement Learning with policy represented via DNN



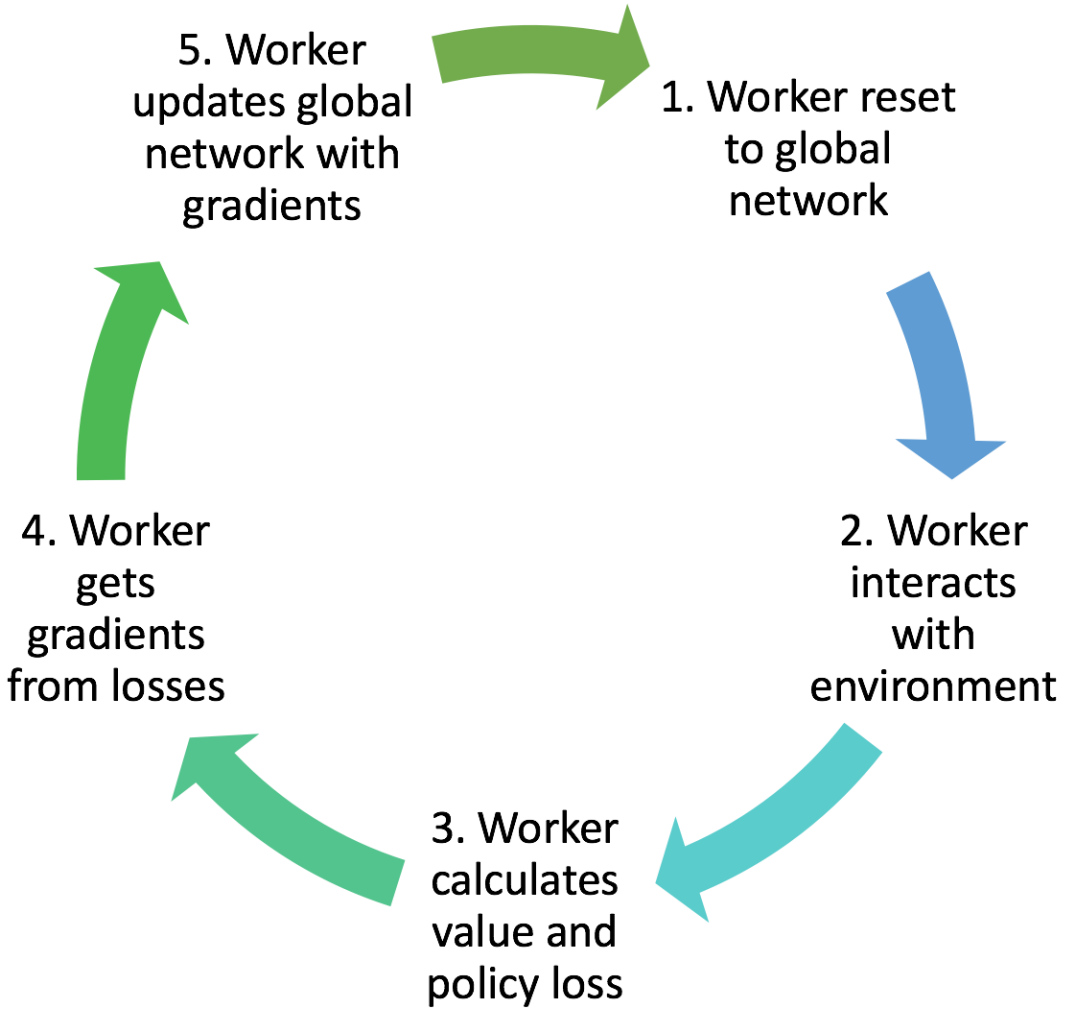
Source: Hongzi Mao, Mohammad Alizadeh, Ishai Menache, and Srikanth Kandula. (2016) "Resource management with deep reinforcement learning." In Proceedings of the 15th ACM Workshop on Hot Topics in Networks, pp. 50-56. ACM, 2016.

Asynchronous Advantage Actor-Critic (A3C)



Source: https://medium.com/emergent-future/simple-reinforcement-learning-with-tensorflow-part-8-asynchronous-actor-critic-agents-a3c-c88f72a5e9f2

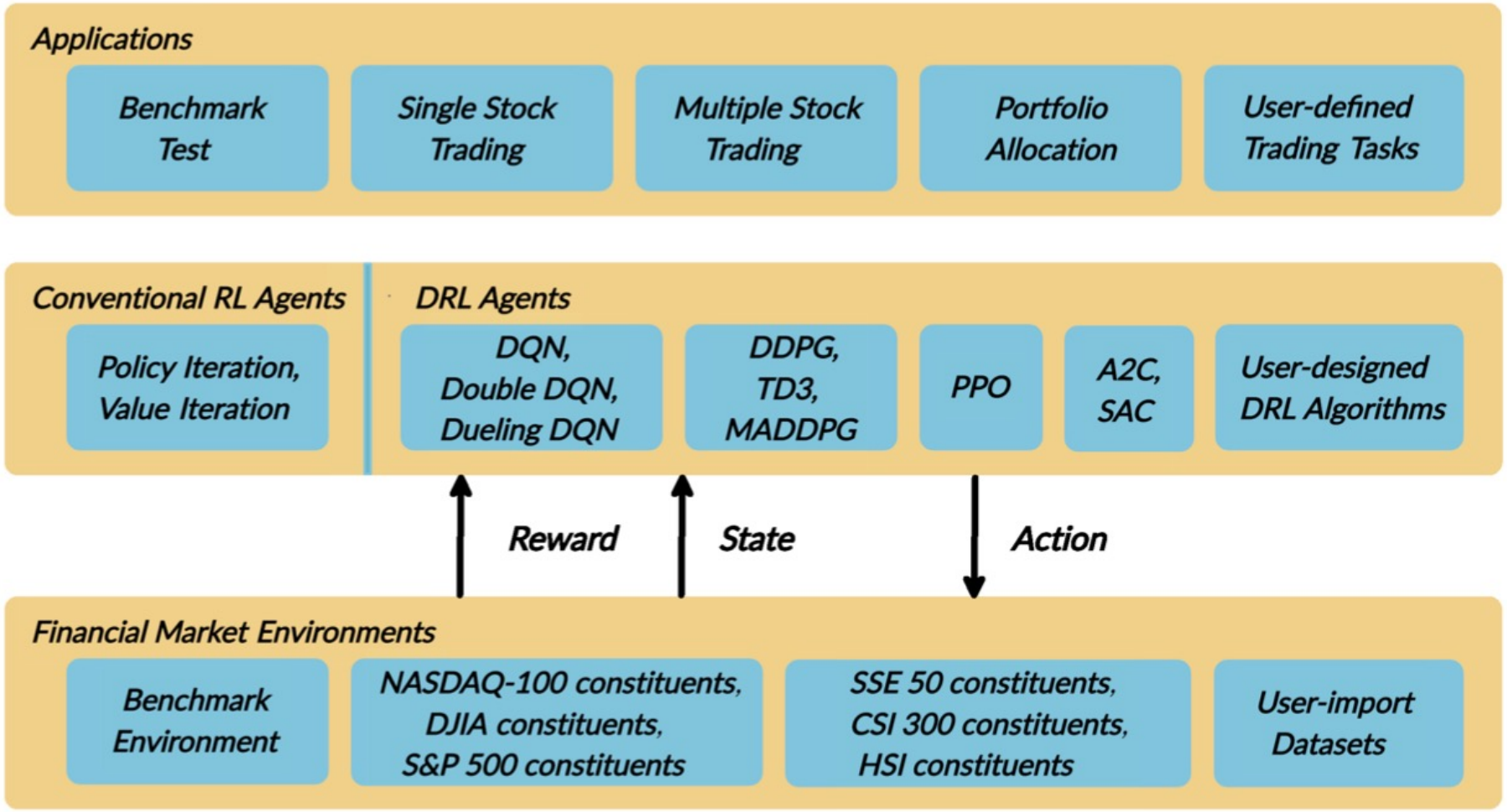
Training workflow of each worker agent in A3C



Source: https://medium.com/emergent-future/simple-reinforcement-learning-with-tensorflow-part-8-asynchronous-actor-critic-agents-a3c-c88f72a5e9f2

FinRL:

A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance



Source: Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang (2020). "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." arXiv preprint arXiv:2011.09607 (2020).

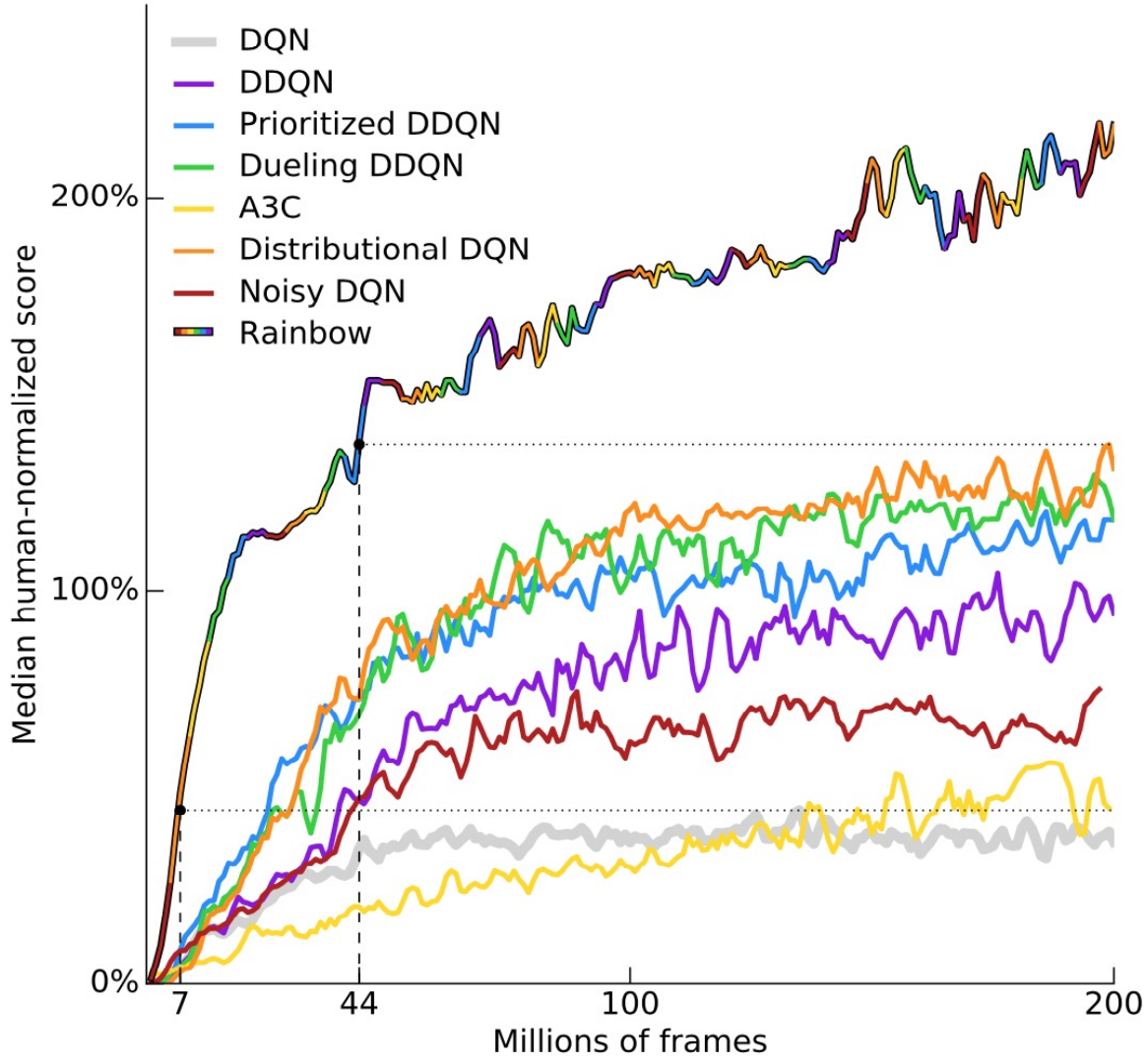
FinRL

Deep Reinforcement Learning Algorithms

Algorithms	Input	Output	Type	State-action spaces support	Finance use cases support	Features and Improvements	Advantages
DQN	States	Q-value	Value based	Discrete only	Single stock trading	Target network, experience replay	Simple and easy to use
Double DQN	States	Q-value	Value based	Discrete only	Single stock trading	Use two identical neural network models to learn	Reduce overestimations
Dueling DQN	States	Q-value	Value based	Discrete only	Single stock trading	Add a specialized dueling Q head	Better differentiate actions, improves the learning
DDPG	State action pair	Q-value	Actor-critic based	Continuous only	Multiple stock trading, portfolio allocation	Being deep Q-learning for continuous action spaces	Better at handling high-dimensional continuous action spaces
A2C	State action pair	Q-value	Actor-critic based	Discrete and continuous	All use cases	Advantage function, parallel gradients updating	Stable, cost-effective, faster and works better with large batch sizes
PPO	State action pair	Q-value	Actor-critic based	Discrete and continuous	All use cases	Clipped surrogate objective function	Improve stability, less variance, simply to implement
SAC	State action pair	Q-value	Actor-critic based	Continuous only	Multiple stock trading, portfolio allocation	Entropy regularization, exploration-exploitation trade-off	Improve stability
TD3	State action pair	Q-value	Actor-critic based	Continuous only	Multiple stock trading, portfolio allocation	Clipped double Q-Learning, delayed policy update, target policy smoothing.	Improve DDPG performance
MADDPG	State action pair	Q-value	Actor-critic based	Continuous only	Multiple stock trading, portfolio allocation	Handle multi-agent RL problem	Improve stability and performance

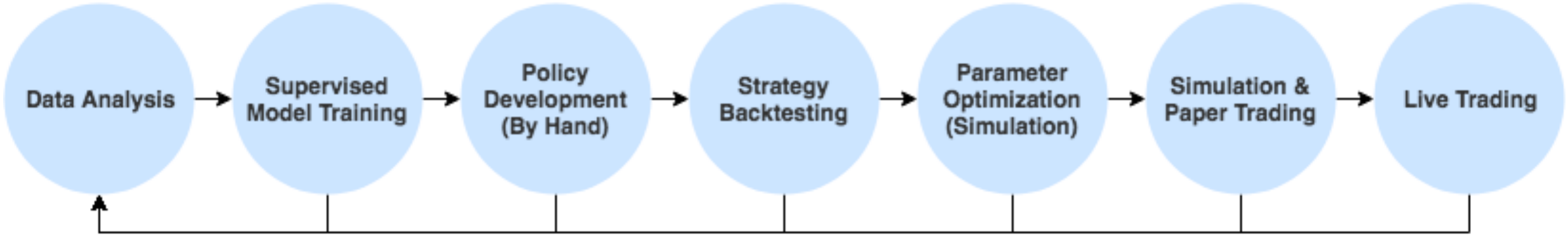
Source: Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang (2020). "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." arXiv preprint arXiv:2011.09607 (2020).

Rainbow: Combining improvements in deep reinforcement learning



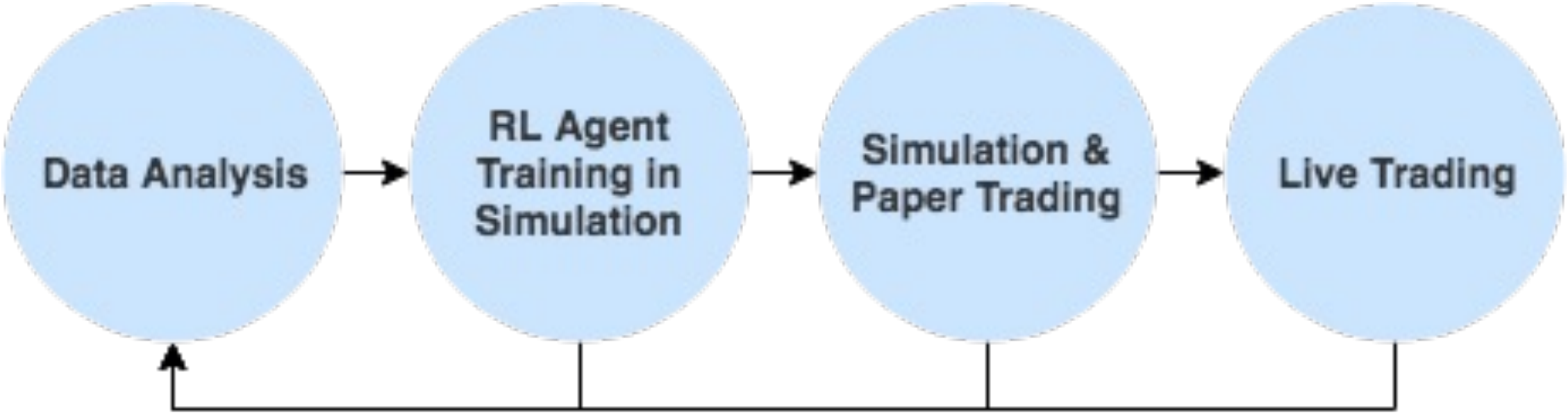
Source: Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver (2017). "Rainbow: Combining improvements in deep reinforcement learning." arXiv preprint arXiv:1710.02298 (2017).

A Typical Strategy Development Workflow



Source: http://www.wildml.com/2018/02/introduction-to-learning-to-trade-with-reinforcement-learning/

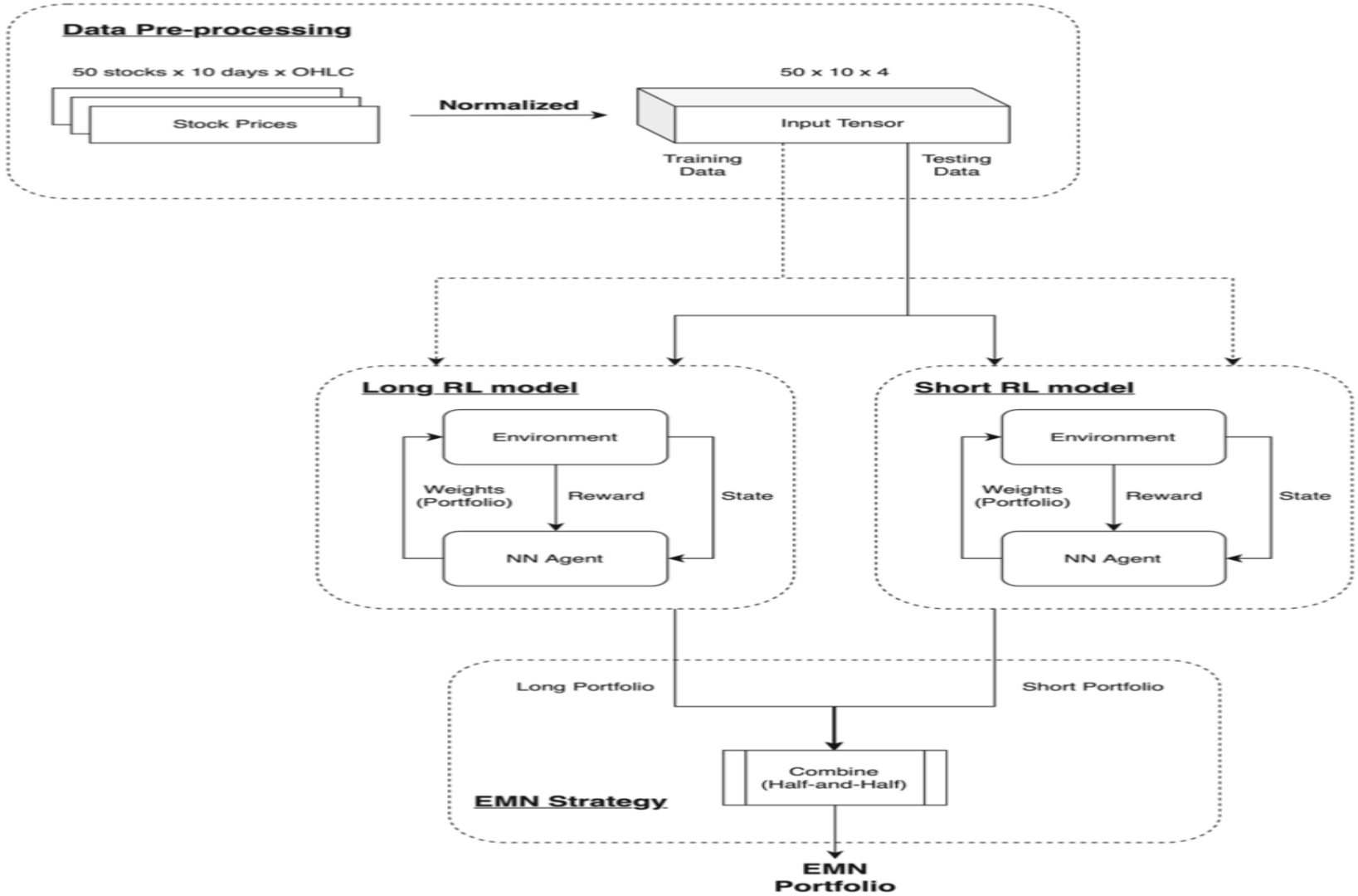
Reinforcement Learning (RL) in Trading Strategies



Source: http://www.wildml.com/2018/02/introduction-to-learning-to-trade-with-reinforcement-learning/

Portfolio management system in equity market neutral using reinforcement learning

(Wu et al., 2021)



Source: Mu-En Wu, Jia-Hao Syu, Jerry Chun-Wei Lin, and Jan-Ming Ho. "Portfolio management system in equity market neutral using reinforcement learning." Applied Intelligence (2021): 1-13.

Deep Reinforcement Learning Library

- OpenAI Gym
- Google Dopamine
- RLlib
- Horizon
- FinRL

Open AI Gym

Environments Documentation

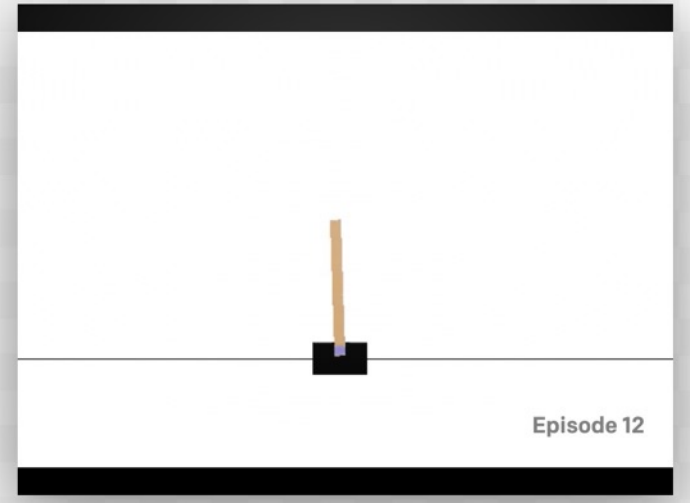
Gym

Gym is a toolkit for developing and comparing reinforcement learning algorithms. It supports teaching agents everything from walking to playing games like Pong or Pinball.

View documentation ›

View on GitHub ›

RandomAgent on Ant-v2



RandomAgent on CartPole-v1

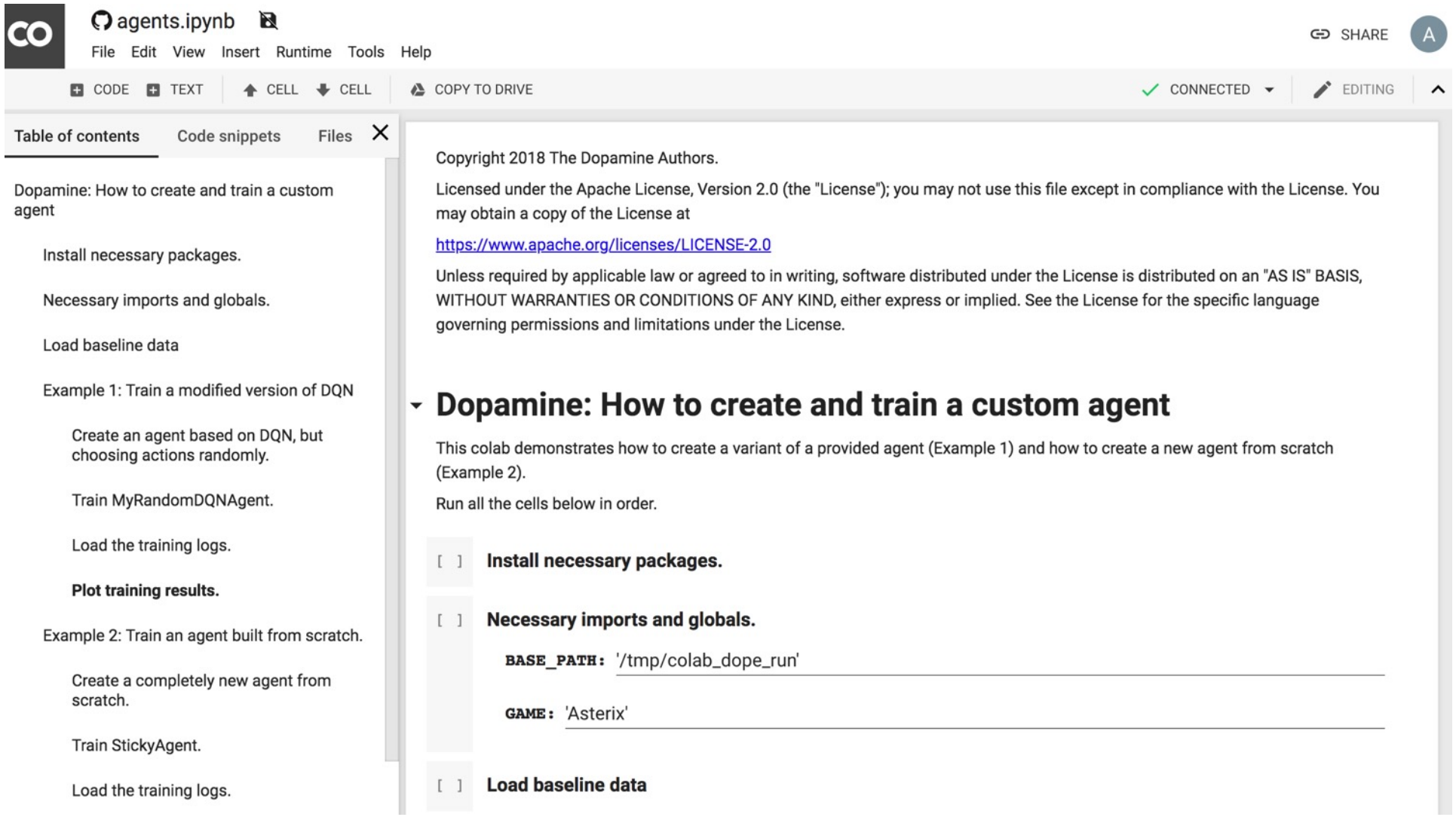
https://gym.openai.com/

Google Dopamine

Dopamine is a research framework
for fast prototyping of
reinforcement learning algorithms.

https://github.com/google/dopamine

Deep Reinforcement Learning
Dopamine Colab Examples
DQN Rainbow

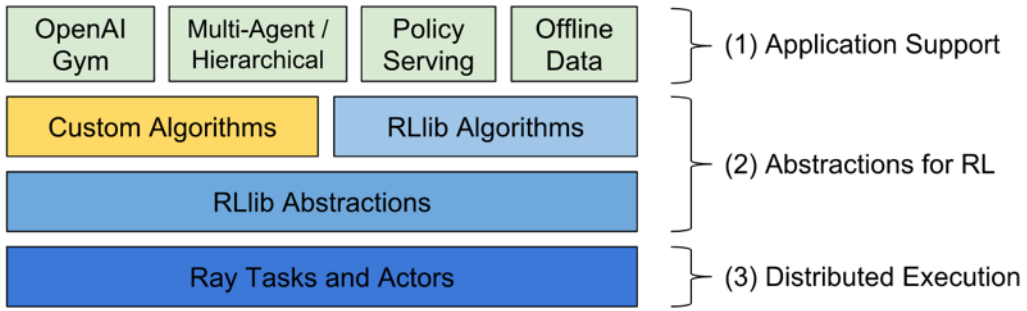


https://colab.research.google.com/github/google/dopamine/blob/master/dopamine/colab/agents.ipynb

RLlib:
Scalable Reinforcement Learning

RLlib: Scalable Reinforcement Learning

RLlib is an open-source library for reinforcement learning that offers both high scalability and a unified API for a variety of applications. RLlib natively supports TensorFlow, TensorFlow Eager, and PyTorch, but most of its internals are framework agnostic.



To get started, take a look over the custom env example and the API documentation. If you're looking to develop custom algorithms with RLlib, also check out concepts and custom algorithms.

RLlib in 60 seconds

The following is a whirlwind overview of RLlib. For a more in-depth guide, see also the full table of contents and RLlib blog posts. You may also want to skim the list of built-in algorithms. Look out for the and icons to see which algorithms are available for each framework.

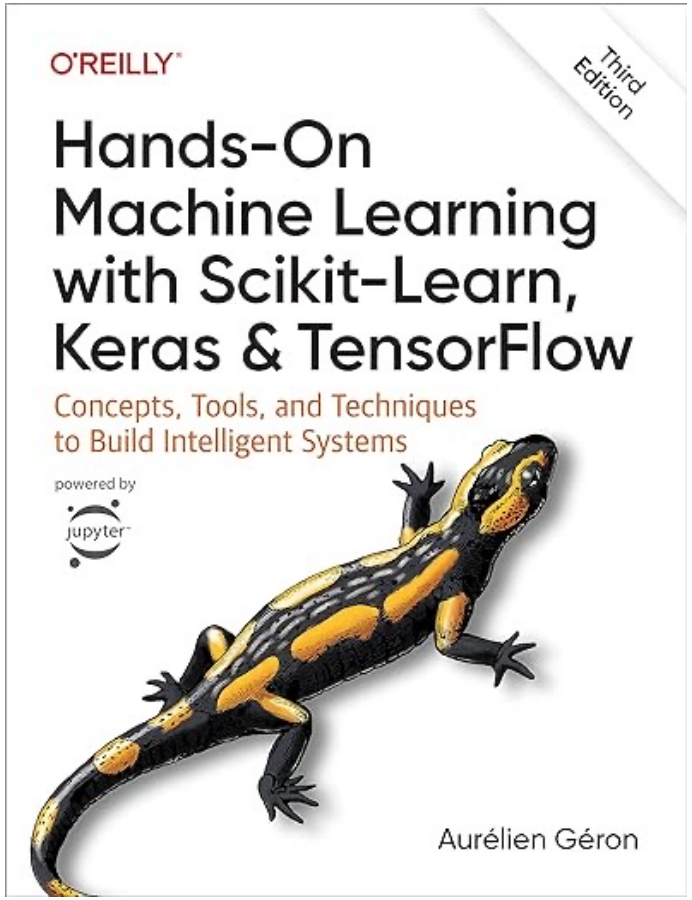
v: master

https://docs.ray.io/en/master/rllib.html

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow

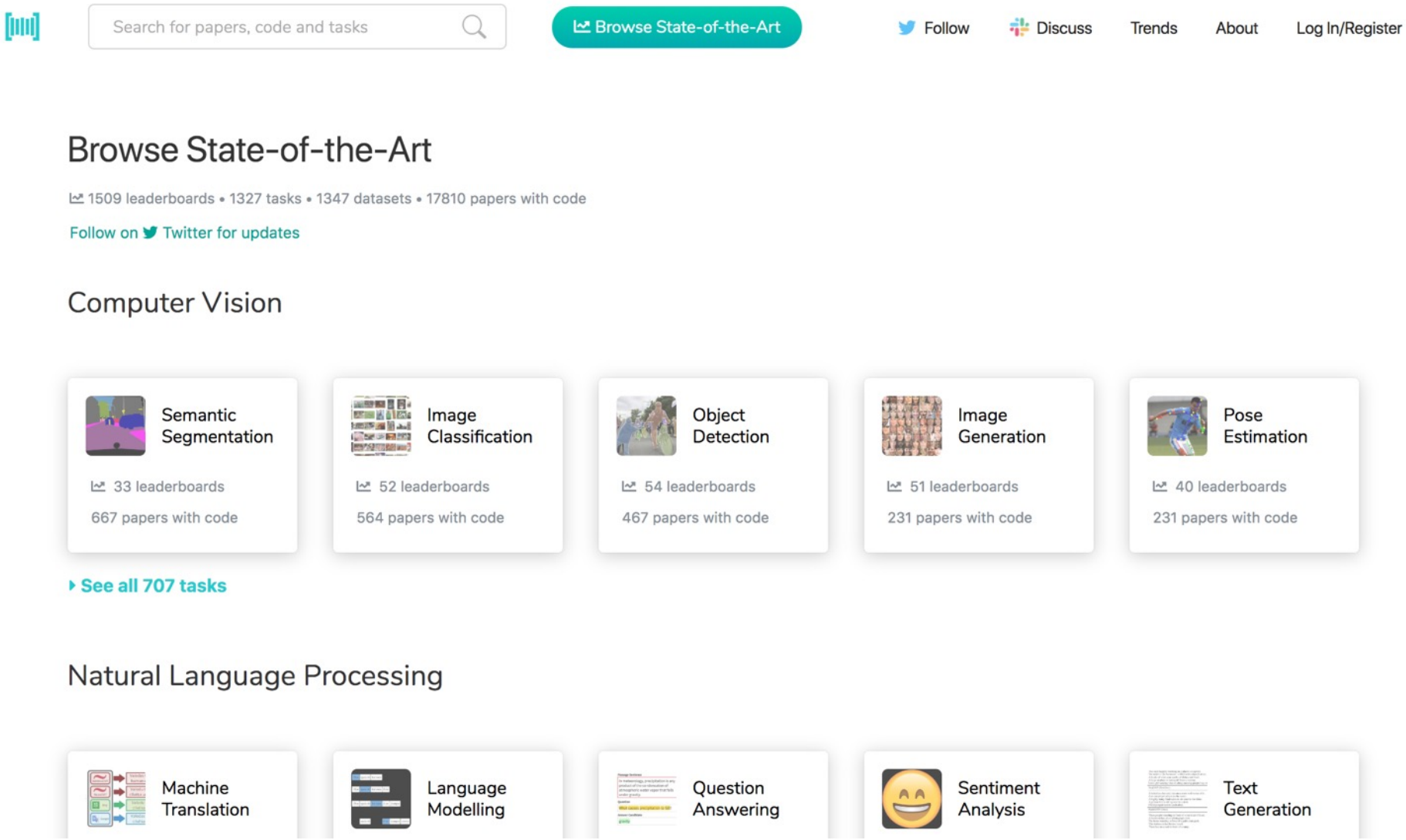
Notebooks

1. The Machine Learning landscape
2. End-to-end Machine Learning project
3. Classification
4. Training Models
5. Support Vector Machines
6. Decision Trees
7. Ensemble Learning and Random Forests
8. Dimensionality Reduction
9. Unsupervised Learning Techniques
10. Artificial Neural Nets with Keras
11. Training Deep Neural Networks
12. Custom Models and Training with TensorFlow
13. Loading and Preprocessing Data
14. **Deep Computer Vision Using Convolutional Neural Networks**
15. **Processing Sequences Using RNNs and CNNs**
16. Natural Language Processing with RNNs and Attention
17. Autoencoders, GANs, and Diffusion Models
18. **Reinforcement Learning**
19. Training and Deploying TensorFlow Models at Scale



https://github.com/ageron/handson-ml3

Papers with Code
State-of-the-Art (SOTA)



https://paperswithcode.com/sota

Summary

- Deep Learning (DL)
  - Neural Networks (NN)
  - Convolutional Neural Networks (CNN)
  - Recurrent Neural Networks (RNN)
- Reinforcement Learning (RL)
  - Markov Decision Processes (MDP)
  - Deep Reinforcement Learning (DRL) Algorithms
    - SARSA
    - Q-Learning
    - DQN, A3C, Rainbow

References

- Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson.
- Tristan Lim (2024). "Environmental, social, and governance (ESG) and artificial intelligence in finance: State-of-the-art and research takeaways." Artificial Intelligence Review, 57, no. 4 (2024): 1-64.
- Min-Yuh Day, Ching-Ying Yang, and Yensen Ni (2024), "Portfolio Dynamic Trading Strategies Using Deep Reinforcement Learning", Soft Computing, Volume 28, August 2024, pp. 8715–8730.
- Numa Dhamani and Maggie Engler (2024), Introduction to Generative AI, Manning
- Denis Rothman (2024), Transformers for Natural Language Processing and Computer Vision - Third Edition: Explore Generative AI and Large Language Models with Hugging Face, ChatGPT, GPT-4V, and DALL-E 3, 3rd ed. Edition, Packt Publishing
- Ben Auffarth (2023), Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT and other LLMs, Packt Publishing.
- Aurélien Géron (2022), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 3rd Edition, O'Reilly Media.
- Steven D'Ascoli (2022), Artificial Intelligence and Deep Learning with Python: Every Line of Code Explained For Readers New to AI and New to Python, Independently published.
- Nithin Buduma, Nikhil Buduma, Joe Papa (2022), Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms, 2nd Edition, O'Reilly Media.
- Aske Plaat, Annie Wong, Suzan Verberne, Joost Broekens, Niki van Stein, and Thomas Back. (2024) "Reasoning with Large Language Models, a Survey." arXiv preprint arXiv:2407.11511.
- Line of Code Explained For Readers New to AI and New to Python, Independently published.
- Nithin Buduma, Nikhil Buduma, Joe Papa (2022), Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms, 2nd Edition, O'Reilly Media.
- Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.
- Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu (2020), "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019." Applied Soft Computing 90 (2020): 106181.
- Deep Learning Basics: Neural Networks Demystified, https://www.youtube.com/playlist?list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU
- Deep Learning SIMPLIFIED, https://www.youtube.com/playlist?list=PLjJh1vlSEYgvGod9wWiydumYl8hOXixNu
- 3Blue1Brown (2017), But what *is* a Neural Network? | Chapter 1, deep learning, https://www.youtube.com/watch?v=aircAruvnKk
- 3Blue1Brown (2017), Gradient descent, how neural networks learn | Chapter 2, deep learning, https://www.youtube.com/watch?v=IHZwWFHWa-w
- 3Blue1Brown (2017), What is backpropagation really doing? | Chapter 3, deep learning, https://www.youtube.com/watch?v=Ilg3gGewQ5U
- Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.
- David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ
- David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, Demis Hassabis (2018), "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play." Science 362, no. 6419 (2018): 1140-1144.
- David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis (2017), "Mastering the game of Go without human knowledge." Nature 550 (2017): 354–359.
- Hado Van Hasselt, Arthur Guez, and David Silver (2016). "Deep Reinforcement Learning with Double Q-Learning." In AAAI, vol. 2, p. 5. 2016.
- Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver (2017). "Rainbow: Combining improvements in deep reinforcement learning." arXiv preprint arXiv:1710.02298 (2017).
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves et al. (2015) "Human-level control through deep reinforcement learning." Nature 518, no. 7540 (2015): 529.
- Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas (2015). "Dueling network architectures for deep reinforcement learning." arXiv preprint arXiv:1511.06581 (2015).
- Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang (2020). "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." arXiv preprint arXiv:2011.09607 (2020).
- Mu-En Wu, Jia-Hao Syu, Jerry Chun-Wei Lin, and Jan-Ming Ho. "Portfolio management system in equity market neutral using reinforcement learning." Applied Intelligence (2021): 1-13.
- Min-Yuh Day (2024), Python 101, https://tinyurl.com/aintpupython101