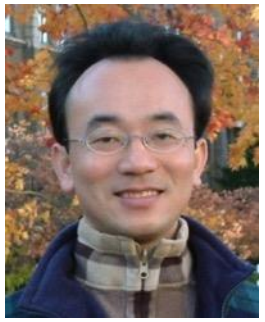
Artificial Intelligence in Finance and Quantitative Analysis

Deep Learning, Reinforcement Learning, and Generative AI in Finance

1132AIFQA08
MBA, IM, NTPU (M5147) (Spring 2025)
Tue 5, 6, 7 (13:10-16:00) (B3F17)

https://meet.google.com/miy-fbif-max



Min-Yuh Day, Ph.D,
Professor

Institute of Information Management, National Taipei University

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

2025-04-29

Syllabus

Week	Date	Subject/Topics
1	2025/02/18	Introduction to Artificial Intelligence in Finance and Quantitative Analysis
2	2025/02/25	AI in FinTech: Metaverse, Web3, DeFi, NFT, Generative AI for Financial Innovation Applications
3	2025/03/04	Investing Psychology and Behavioral Finance
4	2025/03/11	Event Studies in Finance
5	2025/03/18	Case Study on AI in Finance and Quantitative Analysis I
6	2025/03/25	Finance Theory and Data-Driven Finance

Syllabus

Week	Date	Subject/Topics
7	2025/04/01	Self-Study
8	2025/04/08	Midterm Project Report
9	2025/04/15	Financial Econometrics and Machine Learning
10	2025/04/22	AI-First Finance
11	2025/04/29	Deep Learning, Reinforcement Learning, and Generative AI in Finance
12	2025/05/06	Case Study on AI in Finance and Quantitative Analysis II

Syllabus

Week	Date	Subject/Topics
13	2025/05/13	Industry Practices of AI in Finance and Quantitative Analysis
14	2025/05/20	Algorithmic Trading; Risk Management; Trading Bot and Event-Based Backtesting
15	2025/05/27	Final Project Report I
16	2025/06/03	Final Project Report II

Deep Learning in Finance

Reinforcement Learning in Finance

Generative AI in Finance

Outline

- Generative AI in Finance
- Deep Learning (DL) in Finance
  - Dense Neural Networks (DNN)
  - Recurrent Neural Networks (RNN)
  - Convolutional Neural Networks (CNN)
- Reinforcement Learning (RL) in Finance
  - Q Learning (QL)
  - Improved Finance Environment
  - Improved Financial QL Agent

Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Generative AI in Finance

Applications of Generative AI in Finance

- Conversational Finance
- Report Generation
- Financial Query Resolution
- Analysis of Financial Documents
- Synthetic Data Generation
- Stock Behaviour Predictions
- Fraud Detection
- Portfolio Optimisation
- Risk Assessment
- Named Entity Recognition (NER)
- Generating Applicant-Friendly Application Denials

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Applications of Generative AI in Finance

Conversational Finance

Enhances customer service through chatbots and virtual assistants using NLP, providing personalised financial advice, transaction alerts, and payment reminders.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Applications of Generative AI in Finance

Report Generation

Automatically creates financial reports like cash flow statements, income reports, and balance sheets, saving time and reducing human error.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Applications of Generative AI in Finance

Financial Query Resolution

Responds to finance-related queries in real-time using generative models, offering personalised insights for customers and analysts.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Applications of Generative AI in Finance

Analysis of Financial Documents

Applies NLP to automate extraction and summarization of key information from lengthy financial reports and filings.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Applications of Generative AI in Finance

Synthetic Data Generation

Produces artificial financial data for training models, ensuring privacy and overcoming data scarcity while maintaining data realism.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Applications of Generative AI in Finance

Generating Applicant-Friendly Application Denials

Provides clear and understandable reasons for loan or credit application rejections to improve customer transparency and experience.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Applications of Generative AI in Finance

Stock Behaviour Predictions

Uses deep learning (especially RNNs) and NLP to forecast stock price trends and market behavior with higher accuracy than traditional methods.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Applications of Generative AI in Finance

Fraud Detection

Utilises AI to identify suspicious activities, reducing financial risk and preventing fraud by detecting anomalies and unusual transaction patterns.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Applications of Generative AI in Finance

Portfolio Optimisation

Helps in personalising asset management by considering individual risk preferences, expected returns, and financial goals.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Applications of Generative AI in Finance

Risk Assessment

Assesses financial risk using advanced data-driven models to better inform investment and credit decisions.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Applications of Generative AI in Finance

Named Entity Recognition (NER)

Identifies and classifies key financial entities (like company names, stock symbols) from unstructured text to assist in data organization and analysis.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Generative AI for Finance

Aspect	Advantages	Disadvantages
Data Efficiency	Can generate synthetic financial data, reducing the need for large, real datasets.	Synthetic data may not fully capture the complexities of real-world financial data.
Model Complexity	Capable of modelling complex financial systems and patterns.	Complexity makes models hard to interpret, leading to a "black box" issue.
Accuracy	High predictive accuracy for tasks like stock prediction, fraud detection, etc.	High-accuracy models may overfit training data, leading to poor real-world performance.
Adaptability	Can adapt to new financial conditions or types of fraud more quickly.	Adapting models to new events may require expensive re-training.
Anomaly Detection	Excellent at identifying anomalies, which is crucial for fraud detection and risk assessment.	May produce false positives or negatives due to model complexity.
Real-time Analysis	Capable of real-time analysis and decision-making.	Requires heavy computational resources, increasing operational costs.
Operational Costs	Can automate financial tasks, potentially reducing operational costs.	Initial setup and ongoing maintenance can be costly due to high computational requirements.
Ethical Concerns	Can be built with fairness constraints to reduce biased decision-making.	"Black box" models can cause ethical concerns, especially in critical financial decisions.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Generative AI (GAI) in Fraud Detection

Aspect	Existing Methods	Generative AI (GAI)
Data Requirement	Typically requires large, well-labelled datasets for training.	Can generate synthetic data for training, reducing the need for large datasets.
Model Complexity	May use simpler models that might not capture all complexities.	Complex models capable of understanding complicated patterns in the data.
Training Time	Simpler models, hence training is usually faster.	May require longer training time due to complexity.
Accuracy	Accuracy can vary; may not be as high as GAI in some cases.	High degree of accuracy for recognising fraudulent activity.
Adaptability	May require manual feature engineering to adapt to new fraud types.	Can adapt to new types of fraud by retraining the generative model.
Anomaly Detection	Additional algorithms often necessary for anomaly detection.	Excellent at identifying abnormalities, a key feature in fraud detection.
Real-time Analysis	Real-time capabilities may vary; not always suitable.	Capable of real-time fraud detection due to advanced algorithms.
Computational Requirements	Usually lower computational requirements.	Generally higher due to complexity of models.
Regulatory and Ethical Issues	Easier to interpret and explain, beneficial for legal compliance.	May raise concerns due to the "black box" nature of some generative models.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Generative AI for Finance

Product	Description	Key Features	Primary Users
Finance GPT (GPT-F)	A finance-optimized language model trained on financial documents, news, and market data.	Financial research, wealth management, personalized recommendations, risk evaluation, fraud detection.	Financial institutions, analysts.
Bloomberg GPT	Domain-specific model tailored for financial text analysis, integrated into Bloomberg Terminal.	Real-time news summarization, predictive analytics, sentiment analysis, named entity recognition.	Analysts, traders, portfolio managers.
SmartSummaries (AlphaSense)	AI-based market intelligence tool that delivers concise financial summaries and insights from large datasets.	Summarizes filings, news, and reports; detects market sentiment; context-based financial content extraction.	Financial professionals, hedge funds.
Index GPT (JPMorgan)	AI-driven investment advisory service under development, using GPT-like technology to recommend securities and funds to customers.	Investment advice, securities analysis, personalized financial information and suggestions.	Retail investors, financial advisors.

Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Bloomberg GPT

Aspect	Description
Purpose	Designed to analyse financial news and reports.
Data Sources	Financial news, stock market data, and other Bloomberg services.
Primary Users	Financial analysts, traders, portfolio managers, and journalists.
Key Features	Real-time news summarization, predictive analytics for varying market trends.
Natural Language Capabilities	High-level natural language understanding and generation for complex financial topics.
Integration	Effortlessly integrates with Bloomberg Terminal and its other services.
Customization	Allows users to tailor news feeds and alerts based on individual preferences and portfolios.
Security and Compliance	Follows the privacy and data security laws put in place by the financial industry.
Accessibility	Accessible via Bloomberg Terminal, web interface, and mobile applications.
Cost	Mostly a premium service given the specialised financial focus.

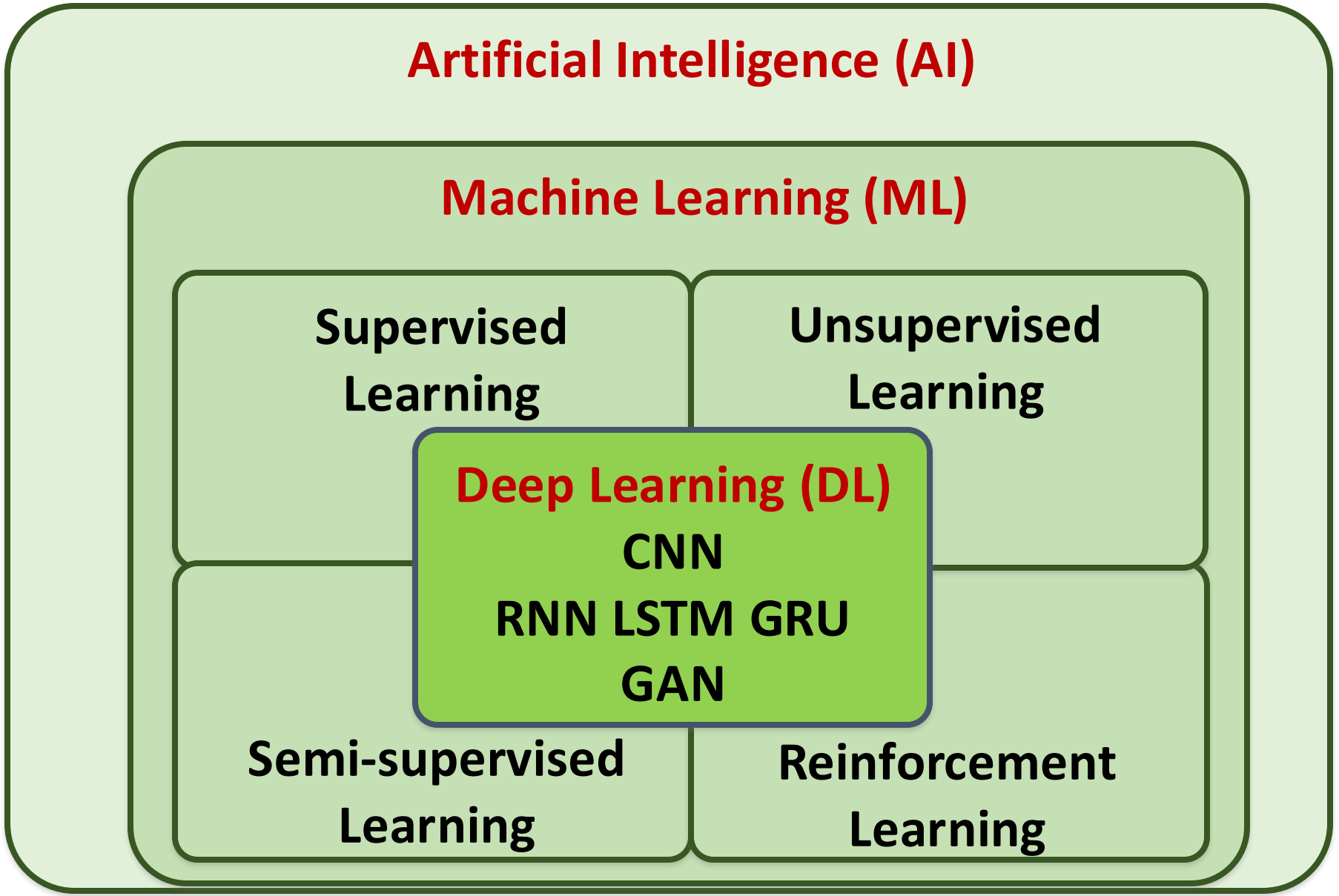
Source: Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.

Deep Learning in Finance

- **Dense Neural Networks (DNN)**
- **Recurrent Neural Networks (RNN)**
- **Convolutional Neural Networks (CNN)**

Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

AI, ML, DL



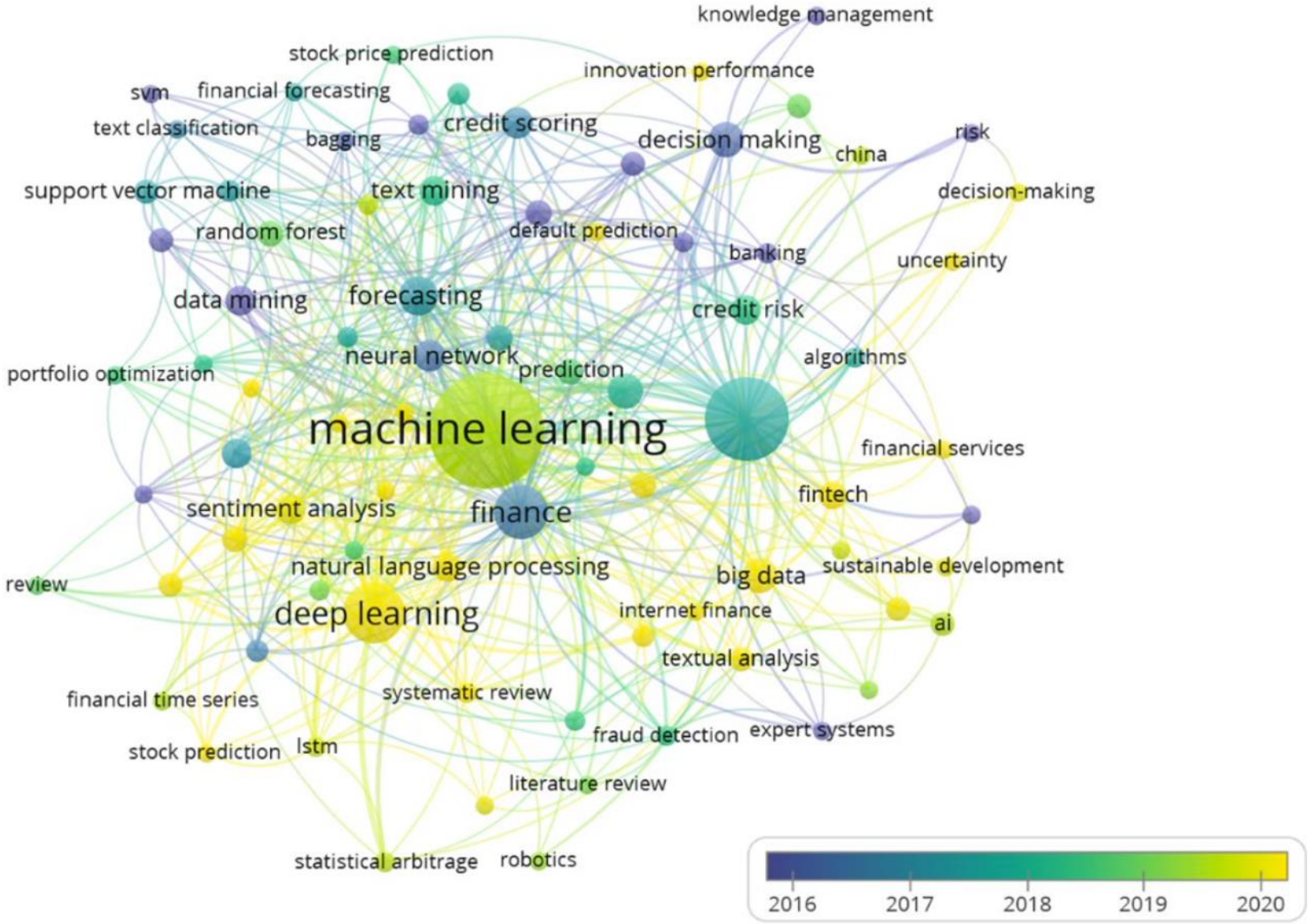
Source: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/deep_learning.html

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

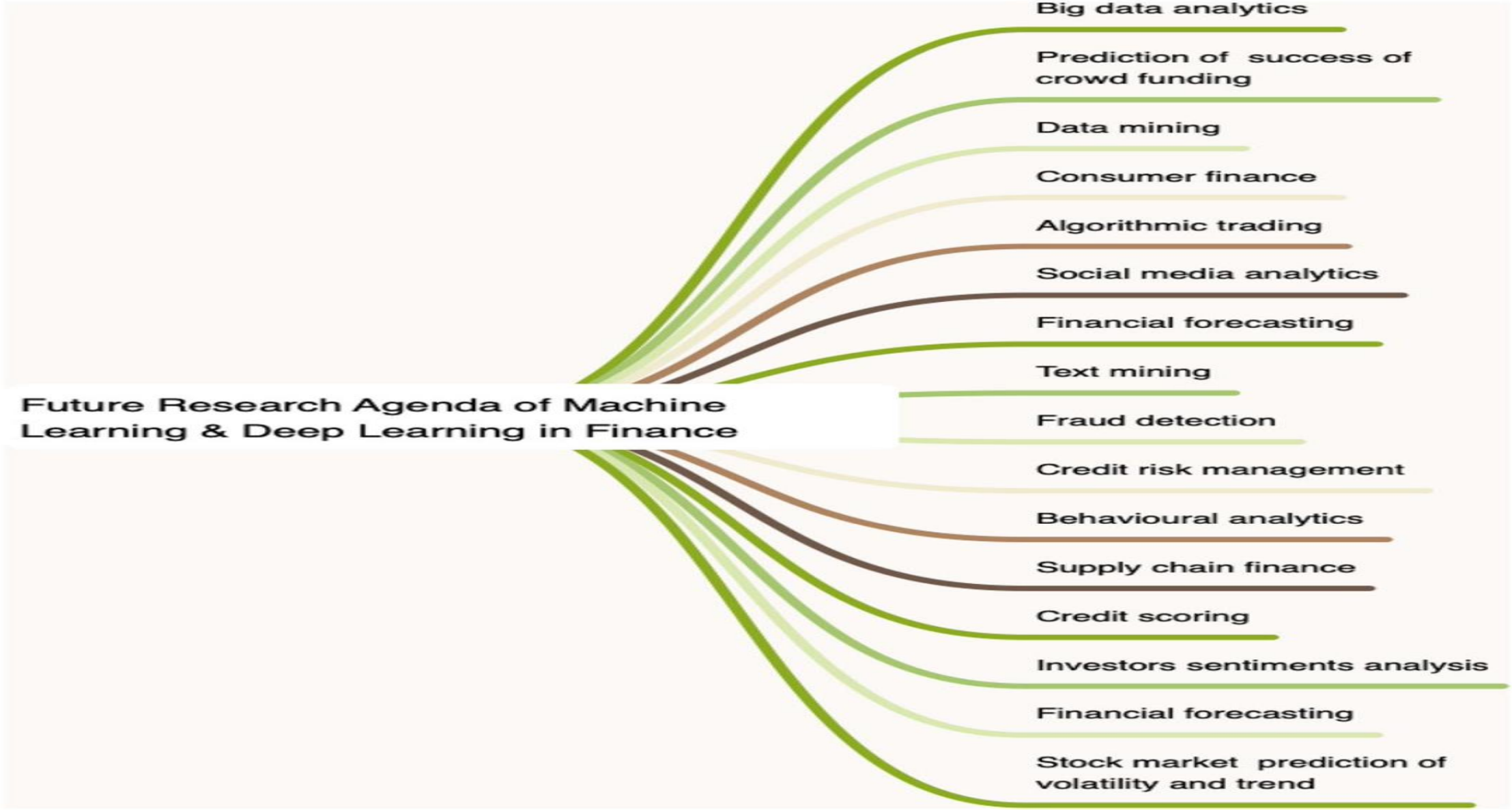


(2016-2022)

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

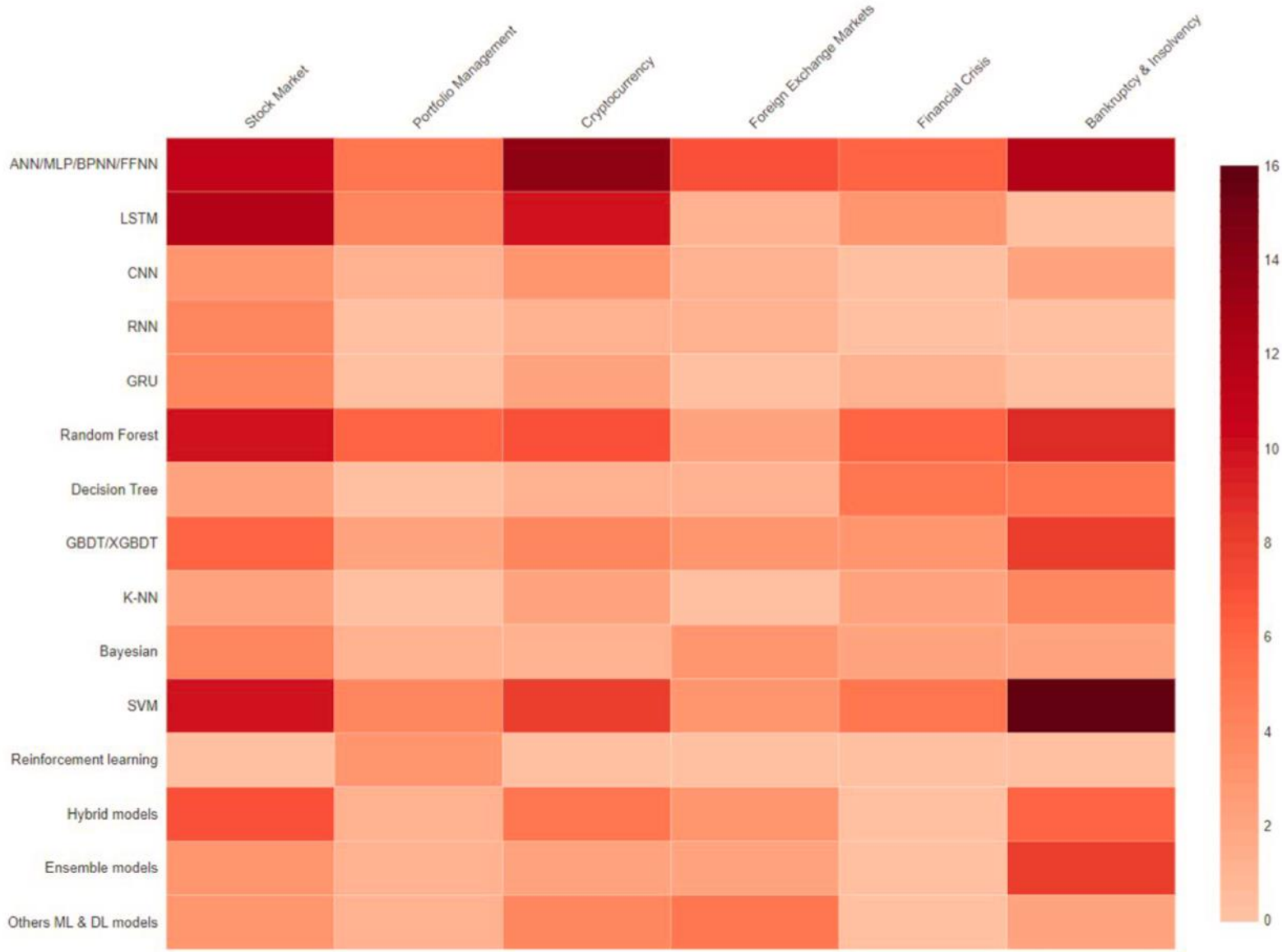


Source: Biju, Ajitha Kumari Vijayappan Nair, Ann Susan Thomas, and J. Thasneem. "Examining the research taxonomy of artificial intelligence, deep learning & machine learning in the financial sphere—a bibliometric analysis." Quality & Quantity (2023): 1-30.

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.

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

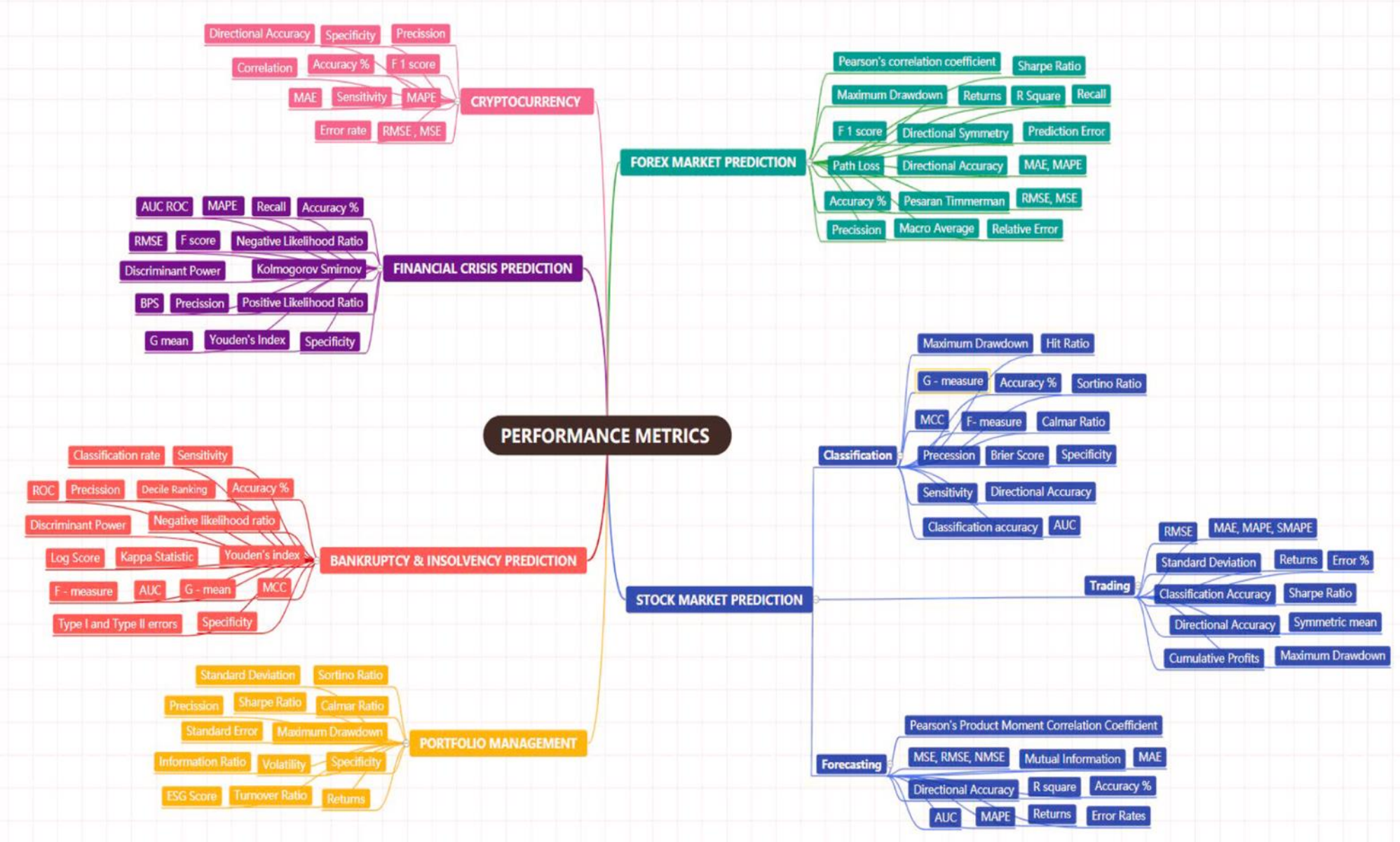
Source: Nazareth, Noella, and Yeruva Yenkata Ramana Reddy. "Financial applications of machine learning: a literature review." Expert Systems with Applications (2023): 119640.

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.

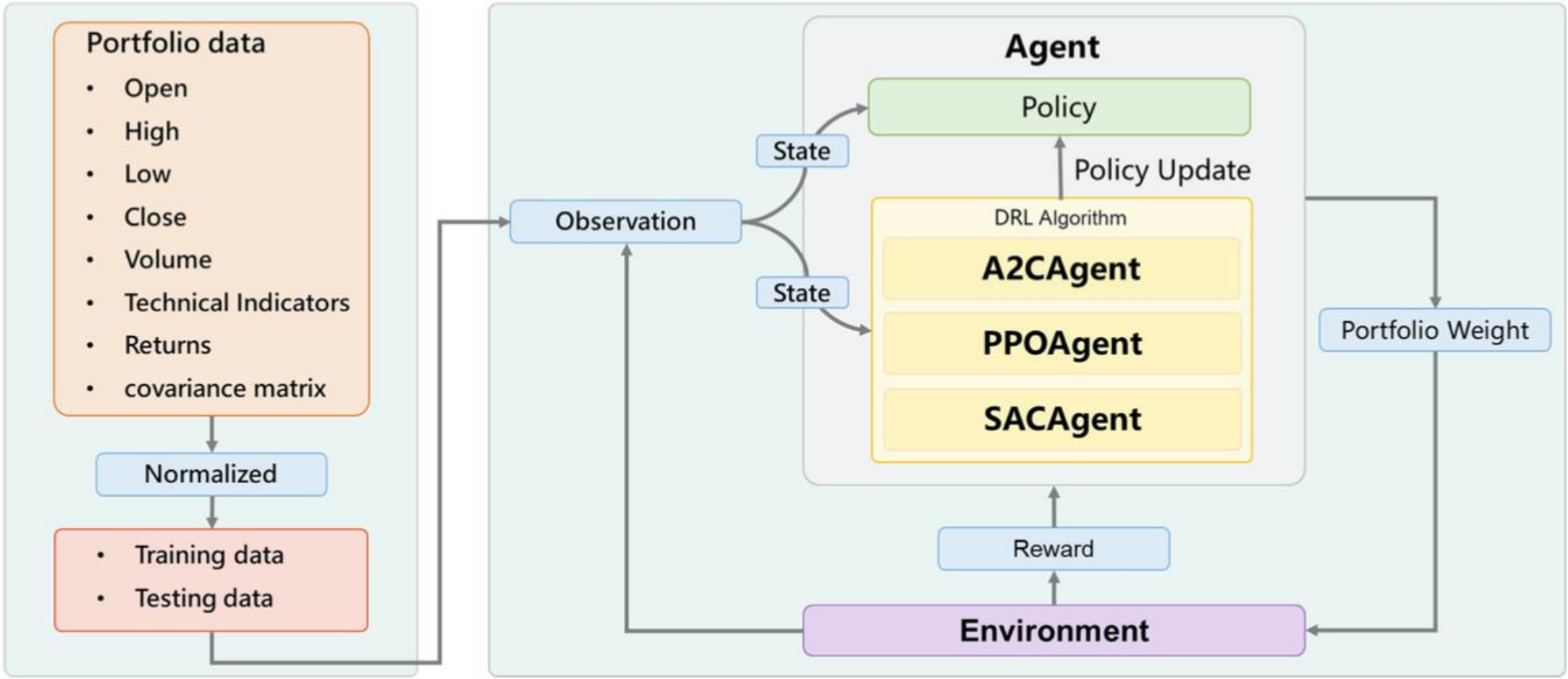
Performance Metrics of Financial Applications of ML



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 for Environmental, Social, and Governance (DRLPMESG)

Day et al. (2023)



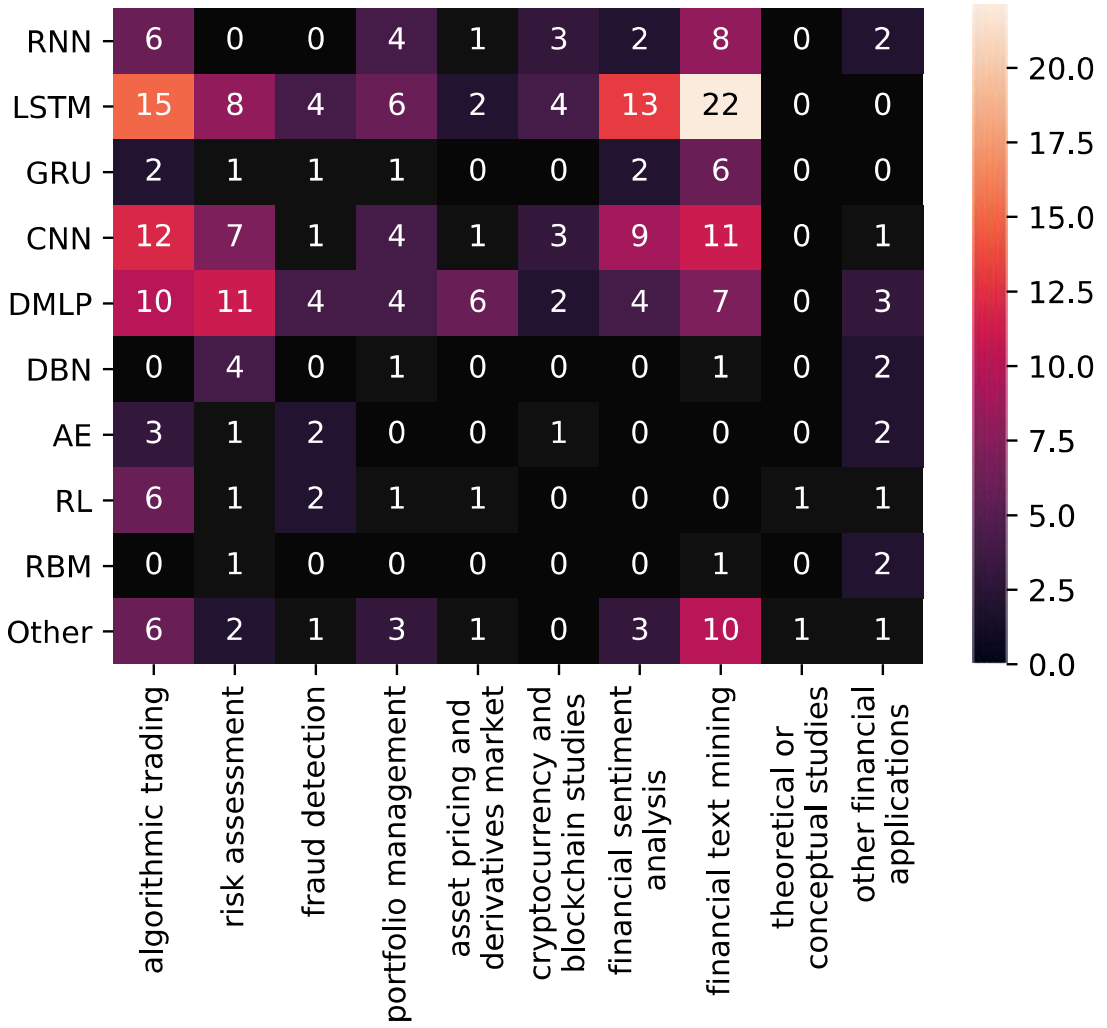
Source: Min-Yuh Day, Ching-Ying Yang, and Yensen Ni (2023), "Portfolio dynamic trading strategies using deep reinforcement learning." Soft Computing (2023): 1-16.

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

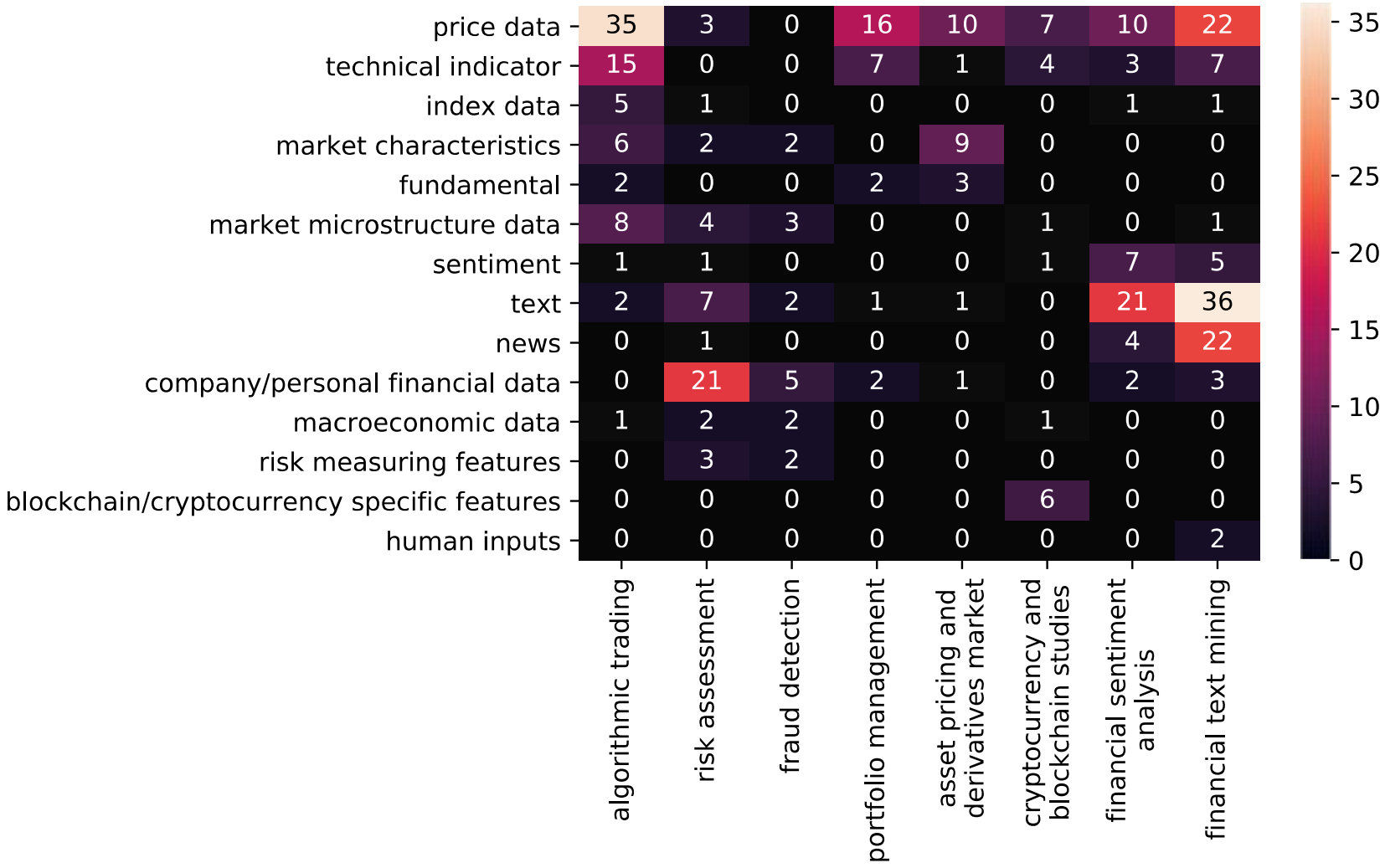
Source: Min-Yuh Day, Ching-Ying Yang, and Yensen Ni (2023), "Portfolio dynamic trading strategies using deep reinforcement learning." Soft Computing (2023): 1-16.

Deep learning for financial applications: Topic-Model Heatmap



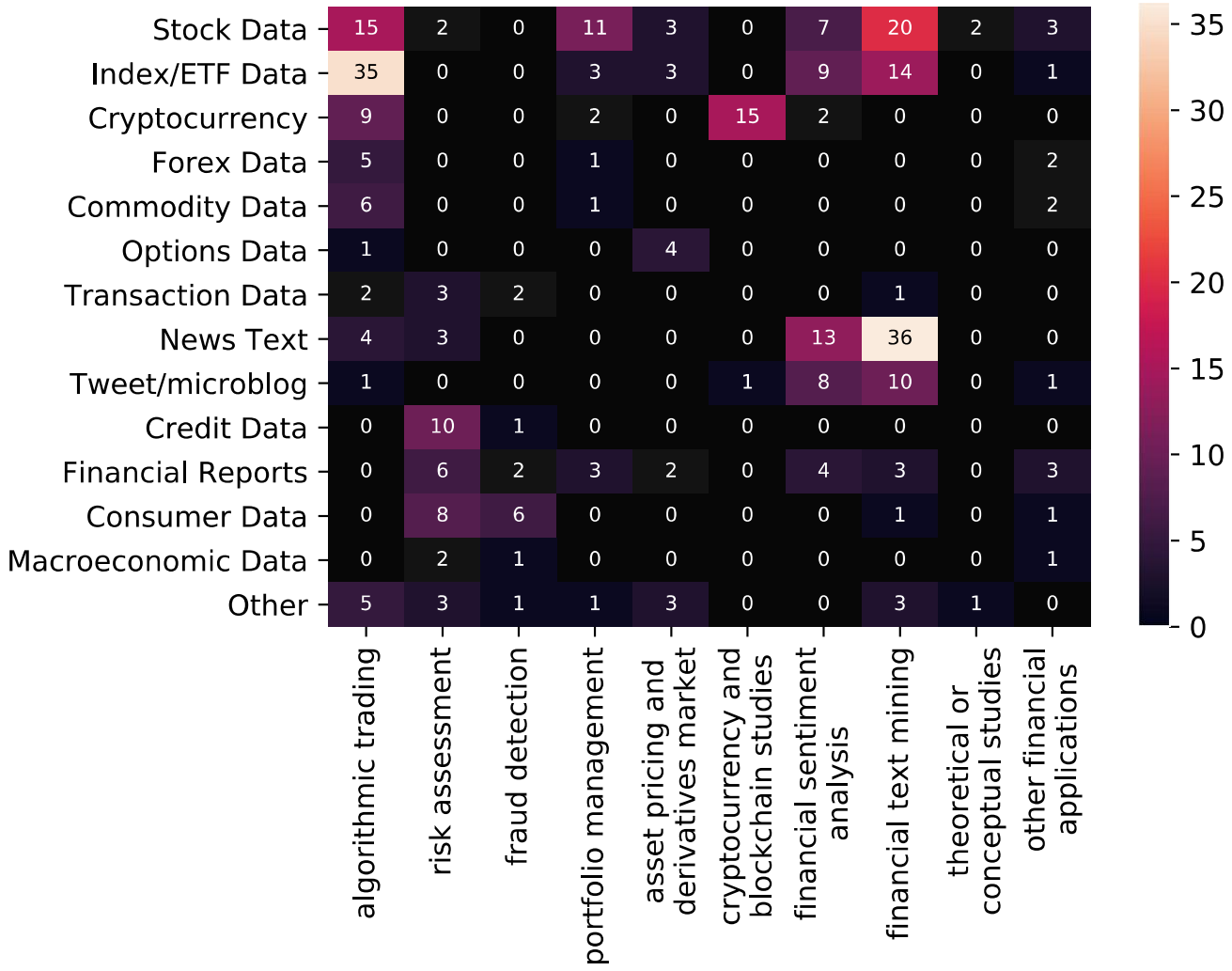
Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications: Topic-Feature Heatmap



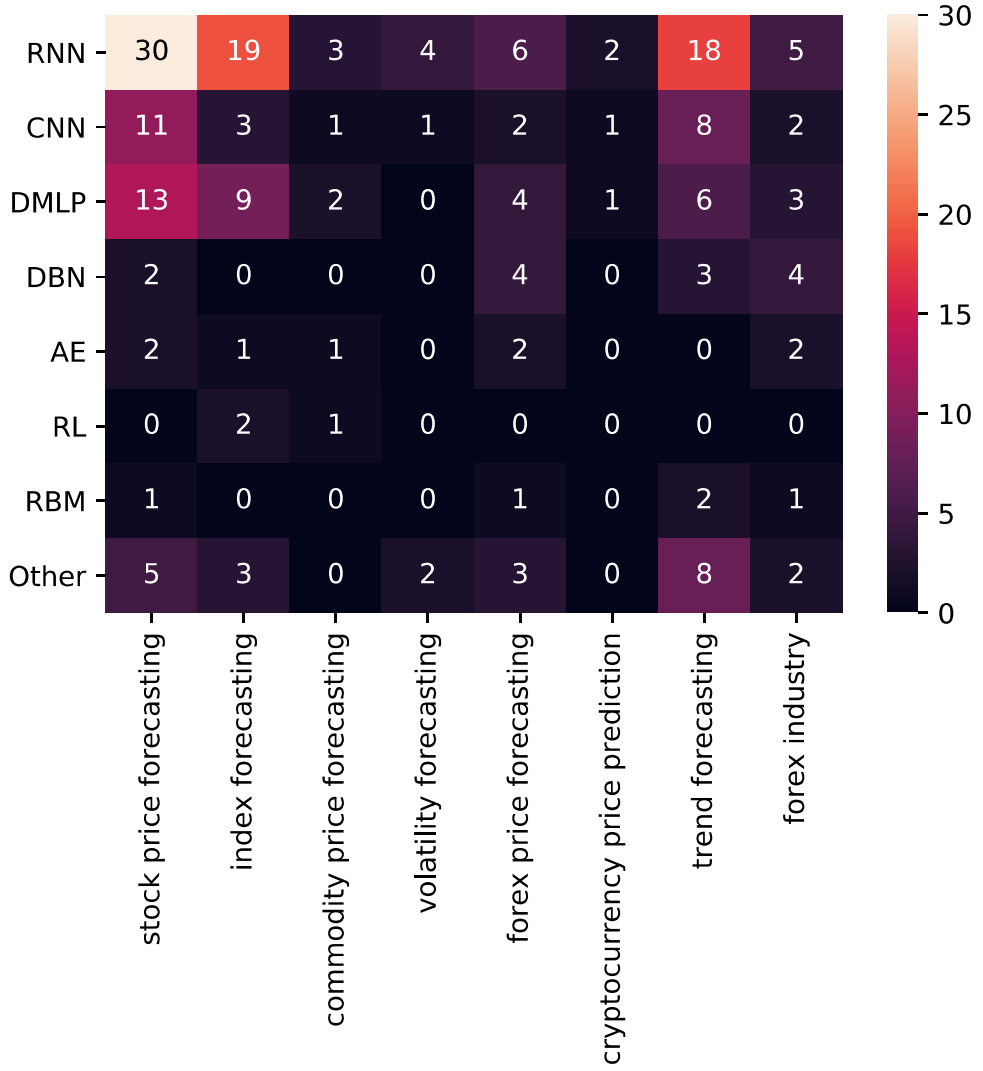
Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications: Topic-Dataset Heatmap



Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Financial time series forecasting with deep learning: Topic-model heatmap



Source: 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 for financial applications:

Algo-trading applications embedded with time series forecasting models

Art.	Data set	Period	Feature set	Method	Performance criteria	Environment
[33]	GarantiBank in BIST, Turkey	2016	OCHLV, Spread, Volatility, Turnover, etc.	PLR, Graves LSTM	MSE, RMSE, MAE, RSE, Correlation R-square	Spark
[34]	CSI300, Nifty50, HSI, Nikkei 225, S&P500, DJIA	2010–2016	OCHLV, Technical Indicators	WT, Stacked autoencoders, LSTM	MAPE, Correlation coefficient, THEIL-U	–
[35]	Chinese Stocks	2007–2017	OCHLV	CNN + LSTM	Annualized Return, Mxm Retracement	Python
[36]	50 stocks from NYSE	2007–2016	Price data	SFM	MSE	–
[37]	The LOB of 5 stocks of Finnish Stock Market	2010	FI-2010 dataset: bid/ask and volume	WMTR, MDA	Accuracy, Precision, Recall, F1-Score	–
[38]	300 stocks from SZSE, Commodity	2014–2015	Price data	FDDR, DMLP+RL	Profit, return, SR, profit-loss curves	Keras
[39]	S&P500 Index	1989–2005	Price data, Volume	LSTM	Return, STD, SR, Accuracy	Python, TensorFlow, Keras, R, H2O
[40]	Stock of National Bank of Greece (ETE).	2009–2014	FTSE100, DJIA, GDAX, NIKKEI225, EUR/USD, Gold	GASVR, LSTM	Return, volatility, SR, Accuracy	Tensorflow
[41]	Chinese stock-IF-IH-IC contract	2016–2017	Decisions for price change	MODRL+LSTM	Profit and loss, SR	–
[42]	Singapore Stock Market Index	2010–2017	OCHL of last 10 days of Index	DMLP	RMSE, MAPE, Profit, SR	–
[43]	GBP/USD	2017	Price data	Reinforcement Learning + LSTM + NES	SR, downside deviation ratio, total profit	Python, Keras, Tensorflow
[44]	Commodity, FX future, ETF	1991–2014	Price Data	DMLP	SR, capability ratio, return	C++, Python
[45]	USD/GBP, S&P500, FTSE100, oil, gold	2016	Price data	AE + CNN	SR, % volatility, avg return/trans, rate of return	H2O

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:

Algo-trading applications embedded with time series forecasting models

Art.	Data set	Period	Feature set	Method	Performance criteria	Environment
[46]	Bitcoin, Dash, Ripple, Monero, Litecoin, Dogecoin, Nxt, Namecoin	2014–2017	MA, BOLL, the CRIX returns, Euribor interest rates, OCHLV	LSTM, RNN, DMLP	Accuracy, F1-measure	Python, Tensorflow
[47]	S&P500, KOSPI, HSI, and EuroStoxx50	1987–2017	200-days stock price	Deep Q-Learning, DMLP	Total profit, Correlation	–
[48]	Stocks in the S&P500	1990–2015	Price data	DMLP, GBT, RF	Mean return, MDD, Calmar ratio	H2O
[49]	Fundamental and Technical Data, Economic Data	–	Fundamental , technical and market information	CNN	–	–

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:

Classification (buy–sell signal, or trend detection) based algo-trading models

Art.	Data set	Period	Feature set	Method	Performance criteria	Environment
[51]	Stocks in Dow30	1997–2017	RSI	DMLP with genetic algorithm	Annualized return	Spark MLlib, Java
[52]	SPY ETF, 10 stocks from S&P500	2014–2016	Price data	FFNN	Cumulative gain	MatConvNet, Matlab
[53]	Dow30 stocks	2012–2016	Close data and several technical indicators	LSTM	Accuracy	Python, Keras, Tensorflow, TALIB
[54]	High-frequency record of all orders	2014–2017	Price data, record of all orders, transactions	LSTM	Accuracy	–
[55]	Nasdaq Nordic (Kesko Oyj, Outokumpu Oyj, Sampo, Rautaruukki, Wartsila Oyj)	2010	Price and volume data in LOB	LSTM	Precision, Recall, F1-score, Cohen's k	–
[56]	17 ETFs	2000–2016	Price data, technical indicators	CNN	Accuracy, MSE, Profit, AUROC	Keras, Tensorflow
[57]	Stocks in Dow30 and 9 Top Volume ETFs	1997–2017	Price data, technical indicators	CNN with feature imaging	Recall, precision, F1-score, annualized return	Python, Keras, Tensorflow, Java
[58]	FTSE100	2000–2017	Price data	CAE	TR, SR, MDD, mean return	–
[59]	Nasdaq Nordic (Kesko Oyj, Outokumpu Oyj, Sampo, Rautaruukki, Wartsila Oyj)	2010	Price, Volume data, 10 orders of the LOB	CNN	Precision, Recall, F1-score, Cohen's k	Theano, Scikit learn, Python
[60]	Borsa Istanbul 100 Stocks	2011–2015	75 technical indicators and OCHLV	CNN	Accuracy	Keras
[61]	ETFs and Dow30	1997–2007	Price data	CNN with feature imaging	Annualized return	Keras, Tensorflow
[62]	8 experimental assets from bond/derivative market	–	Asset prices data	RL, DMLP, Genetic Algorithm	Learning and genetic algorithm error	–
[63]	10 stocks from S&P500	–	Stock Prices	TDNN, RNN, PNN	Missed opportunities, false alarms ratio	–
[64]	London Stock Exchange	2007–2008	Limit order book state, trades, buy/sell orders, order deletions	CNN	Accuracy, kappa	Caffe
[65]	Cryptocurrencies, Bitcoin	2014–2017	Price data	CNN, RNN, LSTM	Accumulative portfolio value, MDD, SR	–

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:

Stand-alone and/or other algorithmic models

Art.	Data set	Period	Feature set	Method	Performance criteria	Environment
[66]	DAX, FTSE100, call/put options	1991–1998	Price data	Markov model, RNN	Ewa-measure, iv, daily profits' mean and std	–
[67]	Taiwan Stock Index Futures, Mini Index Futures	2012–2014	Price data to image	Visualization method + CNN	Accumulated profits,accuracy	–
[68]	Energy-Sector/ Company-Centric Tweets in S&P500	2015–2016	Text and Price data	LSTM, RNN, GRU	Return, SR, precision, recall, accuracy	Python, Tweepy API
[69]	CME FIX message	2016	Limit order book, time-stamp, price data	RNN	Precision, recall, F1-measure	Python, TensorFlow, R
[70]	Taiwan stock index futures (TAIFEX)	2017	Price data	Agent based RL with CNN pre-trained	Accuracy	–
[71]	Stocks from S&P500	2010–2016	OCHLV	DCNL	PCC, DTW, VWL	Pytorch
[72]	News from NowNews, AppleDaily, LTN, MoneyDJ for 18 stocks	2013–2014	Text, Sentiment	DMLP	Return	Python, Tensorflow
[73]	489 stocks from S&P500 and NASDAQ-100	2014–2015	Limit Order Book	Spatial neural network	Cross entropy error	NVIDIA's cuDNN
[74]	Experimental dataset	–	Price data	DRL with CNN, LSTM, GRU, DMLP	Mean profit	Python

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:
Credit scoring or classification studies

Art.	Data set	Period	Feature set	Method	Performance criteria	Env.
[77]	The XR 14 CDS contracts	2016	Recovery rate, spreads, sector and region	DBN+RBM	AUROC, FN, FP, Accuracy	WEKA
[78]	German, Japanese credit datasets	–	Personal financial variables	SVM + DBN	Weighted-accuracy, TP, TN	–
[79]	Credit data from Kaggle	–	Personal financial variables	DMLP	Accuracy, TP, TN, G-mean	–
[80]	Australian, German credit data	–	Personal financial variables	GP + AE as Boosted DMLP	FP	Python, Scikit-learn
[81]	German, Australian credit dataset	–	Personal financial variables	DCNN, DMLP	Accuracy, False/Missed alarm	–
[82]	Consumer credit data from Chinese finance company	–	Relief algorithm chose the 50 most important features	CNN + Relief	AUROC, K-s statistic, Accuracy	Keras
[83]	Credit approval dataset by UCI Machine Learning repo	–	UCI credit approval dataset	Rectifier, Tanh, Maxout DL	–	AWS EC2, H2O, R

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:

Financial distress, bankruptcy, bank risk, mortgage risk, crisis forecasting studies.

Art.	Data set	Period	Feature set	Method	Performance criteria	Env.
[84]	966 french firms	–	Financial ratios	RBM+SVM	Precision, Recall	–
[85]	883 BHC from EDGAR	2006–2017	Tokens, weighted sentiment polarity, leverage and ROA	CNN, LSTM, SVM, RF	Accuracy, Precision, Recall, F1-score	Keras, Python, Scikit-learn
[86]	The event data set for large European banks, news articles from Reuters	2007–2014	Word, sentence	DMLP +NLP preprocess	Relative usefulness, F1-score	–
[87]	Event dataset on European banks, news from Reuters	2007–2014	Text, sentence	Sentence vector + DFFN	Usefulness, F1-score, AUROC	–
[88]	News from Reuters, fundamental data	2007–2014	Financial ratios and news text	doc2vec + NN	Relative usefulness	Doc2vec
[89]	Macro/Micro economic variables, Bank characteristics/performance variables from BHC	1976–2017	Macro economic variables and bank performances	CGAN, MVN, MV-t, LSTM, VAR, FE-QAR	RMSE, Log likelihood, Loan loss rate	–
[90]	Financial statements of French companies	2002–2006	Financial ratios	DBN	Recall, Precision, F1-score, FP, FN	–
[91]	Stock returns of American publicly-traded companies from CRSP	2001–2011	Price data	DBN	Accuracy	Python, Theano
[92]	Financial statements of several companies from Japanese stock market	2002–2016	Financial ratios	CNN	F1-score, AUROC	–
[93]	Mortgage dataset with local and national economic factors	1995–2014	Mortgage related features	DMLP	Negative average log-likelihood	AWS
[94]	Mortgage data from Norwegian financial service group, DNB	2012–2016	Personal financial variables	CNN	Accuracy, Sensitivity, Specificity, AUROC	–
[95]	Private brokerage company's real data of risky transactions	–	250 features: order details, etc.	CNN, LSTM	F1-Score	Keras, Tensorflow
[96]	Several datasets combined to create a new one	1996–2017	Index data, 10-year Bond yield, exchange rates,	Logit, CART, RF, SVM, NN, XGBoost, DMLP	AUROC, KS, G-mean, likelihood ratio, DP, BA, WBA	R

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:

Fraud detection studies

Art.	Data set	Period	Feature set	Method	Performance criteria	Env.
[114]	Debit card transactions by a local Indonesia bank	2016–2017	Financial transaction amount on several time periods	CNN, Stacked-LSTM, CNN-LSTM	AUROC	–
[115]	Credit card transactions from retail banking	2017	Transaction variables and several derived features	LSTM, GRU	Accuracy	Keras
[116]	Card purchases' transactions	2014–2015	Probability of fraud per currency/origin country, other fraud related features	DMLP	AUROC	–
[117]	Transactions made with credit cards by European cardholders	2013	Personal financial variables to PCA	DMLP, RF	Recall, Precision, Accuracy	–
[118]	Credit-card transactions	2015	Transaction and bank features	LSTM	AUROC	Keras, Scikit-learn
[119]	Databases of foreign trade of the Secretariat of Federal Revenue of Brazil	2014	8 Features: Foreign Trade, Tax, Transactions, Employees, Invoices, etc	AE	MSE	H2O, R
[120]	Chamber of Deputies open data, Companies data from Secretariat of Federal Revenue of Brazil	2009–2017	21 features: Brazilian State expense, party name, Type of expense, etc.	Deep Autoencoders	MSE, RMSE	H2O, R
[121]	Real-world data for automobile insurance company labeled as fradulent	–	Car, insurance and accident related features	DMLP + LDA	TP, FP, Accuracy, Precision, F1-score	–
[122]	Transactions from a giant online payment platform	2006	Personal financial variables	GBDT+DMLP	AUROC	–
[123]	Financial transactions	–	Transaction data	LSTM	t-SNE	–
[124]	Empirical data from Greek firms	–	–	DQL	Revenue	Torch

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:

Portfolio management studies

Art.	Data set	Period	Feature set	Method	Performance criteria	Env.
[65]	Cryptocurrencies, Bitcoin	2014–2017	Price data	CNN, RNN, LSTM	Accumulative portfolio value, MDD, SR	–
[127]	Stocks from NYSE, AMEX, NASDAQ	1965–2009	Price data	Autoencoder + RBM	Accuracy, confusion matrix	–
[128]	20 stocks from S&P500	2012–2015	Technical indicators	DMLP	Accuracy	Python, Scikit Learn, Keras, Theano
[129]	Chinese stock data	2012–2013	Technical, fundamental data	Logistic Regression, RF, DMLP	AUC, accuracy, precision, recall, f1, tpr, fpr	Keras, Tensorflow, Python, Scikit learn
[130]	Top 5 companies in S&P500	–	Price data and Financial ratios	LSTM, Auto-encoding, Smart indexing	CAGR	–
[131]	IBB biotechnology index, stocks	2012–2016	Price data	Auto-encoding, Calibrating, Validating, Verifying	Returns	–
[132]	Taiwans stock market	–	Price data	Elman RNN	MSE, return	–
[133]	FOREX (EUR/USD, etc.), Gold	2013	Price data	Evolino RNN	Return	Python
[134]	Stocks in NYSE, AMEX, NASDAQ, TAQ intraday trade	1993–2017	Price, 15 firm characteristics	LSTM+DMLP	Monthly return, SR	Python,Keras, Tensorflow in AWS
[135]	S&P500	1985–2006	monthly and daily log-returns	DBN+MLP	Validation, Test Error	Theano, Python, Matlab
[136]	10 stocks in S&P500	1997–2016	OCHLV, Price data	RNN, LSTM, GRU	Accuracy, Monthly return	Keras, Tensorflow
[137]	Analyst reports on the TSE and Osaka Exchange	2016–2018	Text	LSTM, CNN, Bi-LSTM	Accuracy, R^2	R, Python, MeCab
[138]	Stocks from Chinese/American stock market	2015–2018	OCHLV, Fundamental data	DDPG, PPO	SR, MDD	–
[139]	Hedge fund monthly return data	1996–2015	Return, SR, STD, Skewness, Kurtosis, Omega ratio, Fund alpha	DMLP	Sharpe ratio, Annual return, Cum. return	–
[140]	12 most-volumed cryptocurrency	2015–2016	Price data	CNN + RL	SR, portfolio value, MDD	–

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:

Asset pricing and derivatives market studies

Art.	Der. type	Data set	Period	Feature set	Method	Performance criteria	Env.
[137]	Asset pricing	Analyst reports on the TSE and Osaka Exchange	2016–2018	Text	LSTM, CNN, Bi-LSTM	Accuracy, R^2	R, Python, MeCab
[142]	Options	Simulated a range of call option prices	–	Price data, option strike/maturity, dividend/risk free rates, volatility	DMLP	RMSE, the average percentage pricing error	Tensorflow
[143]	Futures, Options	TAIEX Options	2017	OCHLV, fundamental analysis, option price	DMLP, DMLP with Black scholes	RMSE, MAE, MAPE	–
[144]	Equity returns	Returns in NYSE, AMEX, NASDAQ	1975–2017	57 firm characteristics	Fama–French n-factor model DL	R^2,RMSE	Tensorflow

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:
Cryptocurrency and blockchain studies

Art.	Data set	Period	Feature set	Method	Performance criteria	Env.
[46]	Bitcoin, Dash, Ripple, Monero, Litecoin, Dogecoin, Nxt, Namecoin	2014–2017	MA, BOLL, the CRIX daily returns, Euribor interest rates, OCHLV of EURO/UK, EURO/USD, US/JPY	LSTM, RNN, DMLP	Accuracy, F1-measure	Python, Tensorflow
[65]	Cryptocurrencies, Bitcoin	2014–2017	Price data	CNN	Accumulative portfolio value, MDD, SR	–
[140]	12 most-volumed cryptocurrency	2015–2016	Price data	CNN + RL	SR, portfolio value, MDD	
[145]	Bitcoin data	2010–2017	Hash value, bitcoin address, public/private key, digital signature, etc.	Takagi–Sugeno Fuzzy cognitive maps	Analytical hierarchy process	–
[146]	Bitcoin data	2012, 2013, 2016	TransactionId, input/output Addresses, timestamp	Graph embedding using heuristic, laplacian eigen-map, deep AE	F1-score	–
[147]	Bitcoin, Litecoin, StockTwits	2015–2018	OCHLV, technical indicators, sentiment analysis	CNN, LSTM, State Frequency Model	MSE	Keras, Tensorflow
[148]	Bitcoin	2013–2016	Price data	Bayesian optimized RNN, LSTM	Sensitivity, specificity, precision, accuracy, RMSE	Keras, Python, Hyperas

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:

Financial sentiment studies coupled with text mining for forecasting

Art.	Data set	Period	Feature set	Method	Performance criteria	Env.
[137]	Analyst reports on the TSE and Osaka Exchange	2016–2018	Text	LSTM, CNN, Bi-LSTM	Accuracy, R^2	R, Python, MeCab
[150]	Sina Weibo, Stock market records	2012–2015	Technical indicators, sentences	DRSE	F1-score, precision, recall, accuracy, AUROC	Python
[151]	News from Reuters and Bloomberg for S&P500 stocks	2006–2015	Financial news, price data	DeepClue	Accuracy	Dynet software
[152]	News from Reuters and Bloomberg, Historical stock security data	2006–2013	News, price data	DMLP	Accuracy	–
[153]	SCI prices	2008–2015	OCHL of change rate, price	Emotional Analysis + LSTM	MSE	–
[154]	SCI prices	2013–2016	Text data and Price data	LSTM	Accuracy, F1-Measure	Python, Keras
[155]	Stocks of Google, Microsoft and Apple	2016–2017	Twitter sentiment and stock prices	RNN	–	Spark, Flume,Twitter API,
[156]	30 DJIA stocks, S&P500, DJI, news from Reuters	2002–2016	Price data and features from news articles	LSTM, NN, CNN and word2vec	Accuracy	VADER
[157]	Stocks of CSI300 index, OCHLV of CSI300 index	2009–2014	Sentiment Posts, Price data	Naive Bayes + LSTM	Precision, Recall, F1-score, Accuracy	Python, Keras
[158]	S&P500, NYSE Composite, DJIA, NASDAQ Composite	2009–2011	Twitter moods, index data	DNN, CNN	Error rate	Keras, Theano

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:

Text mining studies without sentiment analysis for forecasting

Art.	Data set	Period	Feature set	Method	Performance criteria	Env.
[68]	Energy-Sector/ Company-Centric Tweets in S&P500	2015–2016	Text and Price data	RNN, KNN, SVR, LinR	Return, SR, precision, recall, accuracy	Python, Tweepy API
[165]	News from Reuters, Bloomberg	2006–2013	Financial news, price data	Bi-GRU	Accuracy	Python, Keras
[166]	News from Sina.com, ACE2005 Chinese corpus	2012–2016	A set of news text	Their unique algorithm	Precision, Recall, F1-score	–
[167]	CDAX stock market data	2010–2013	Financial news, stock market data	LSTM	MSE, RMSE, MAE, Accuracy, AUC	TensorFlow, Theano, Python, Scikit-Learn
[168]	Apple, Airbus, Amazon news from Reuters, Bloomberg, S&P500 stock prices	2006–2013	Price data, news, technical indicators	TGRU, stock2vec	Accuracy, precision, AUROC	Keras, Python
[169]	S&P500 Index, 15 stocks in S&P500	2006–2013	News from Reuters and Bloomberg	CNN	Accuracy, MCC	–
[170]	S&P500 index news from Reuters	2006–2013	Financial news titles, Technical indicators	SI-RCNN (LSTM + CNN)	Accuracy	–
[171]	10 stocks in Nikkei 225 and news	2001–2008	Textual information and Stock prices	Paragraph Vector + LSTM	Profit	–
[172]	NIFTY50 Index, NIFTY Bank/Auto/IT/Energy Index, News	2013–2017	Index data, news	LSTM	MCC, Accuracy	–
[173]	Price data, index data, news, social media data	2015	Price data, news from articles and social media	Coupled matrix and tensor	Accuracy, MCC	Jieba
[174]	HS300	2015–2017	Social media news, price data	RNN-Boost with LDA	Accuracy, MAE, MAPE, RMSE	Python, Scikit-learn

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:

Text mining studies without sentiment analysis for forecasting

[175]	News and Chinese stock data	2014–2017	Selected words in a news	HAN	Accuracy, Annual return	–
[176]	News, stock prices from Hong Kong Stock Exchange	2001	Price data and TF-IDF from news	ELM, DLR, PCA, BELM, KELM, NN	Accuracy	Matlab
[177]	TWSE index, 4 stocks in TWSE	2001–2017	Technical indicators, Price data, News	CNN + LSTM	RMSE, Profit	Keras, Python, TALIB
[178]	Stock of Tsugami Corporation	2013	Price data	LSTM	RMSE	Keras, Tensorflow
[179]	News, Nikkei Stock Average and 10-Nikkei companies	1999–2008	news, MACD	RNN, RBM+DBN	Accuracy, *P*-value	–
[180]	ISMIS 2017 Data Mining Competition dataset	–	Expert identifier, classes	LSTM + GRU + FFNN	Accuracy	–
[181]	Reuters, Bloomberg News, S&P500 price	2006–2013	News and sentences	LSTM	Accuracy	–
[182]	APPL from S&P500 and news from Reuters	2011–2017	Input news, OCHLV, Technical indicators	CNN + LSTM, CNN+SVM	Accuracy, F1-score	Tensorflow
[183]	Nikkei225, S&P500, news from Reuters and Bloomberg	2001–2013	Stock price data and news	DGM	Accuracy, MCC, %profit	–
[184]	Stocks from S&P500	2006–2013	Text (news) and Price data	LAR+News, RF+News	MAPE, RMSE	–

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:

Financial sentiment studies coupled with text mining without forecasting

Art.	Data set	Period	Feature set	Method	Performance criteria	Env.
[85]	883 BHC from EDGAR	2006–2017	Tokens, weighted sentiment polarity, leverage and ROA	CNN, LSTM, SVM, Random Forest	Accuracy, Precision, Recall, F1-score	Keras, Python, Scikit-learn
[185]	SemEval-2017 dataset, financial text, news, stock market data	2017	Sentiments in Tweets, News headlines	Ensemble SVR, CNN, LSTM, GRU	Cosine similarity score, agreement score, class score	Python, Keras, Scikit Learn
[186]	Financial news from Reuters	2006–2015	Word vector, Lexical and Contextual input	Targeted dependency tree LSTM	Cumulative abnormal return	–
[187]	Stock sentiment analysis from StockTwits	2015	StockTwits messages	LSTM, Doc2Vec, CNN	Accuracy, precision, recall, f-measure, AUC	–
[188]	Sina Weibo, Stock market records	2012–2015	Technical indicators, sentences	DRSE	F1-score, precision, recall, accuracy, AUROC	Python
[189]	News from NowNews, AppleDaily, LTN, MoneyDJ for 18 stocks	2013–2014	Text, Sentiment	LSTM, CNN	Return	Python, Tensorflow
[190]	StockTwits	2008–2016	Sentences, StockTwits messages	CNN, LSTM, GRU	MCC, WSURT	Keras, Tensorflow
[191]	Financial statements of Japan companies	–	Sentences, text	DMLP	Precision, recall, f-score	–
[192]	Twitter posts, news headlines	–	Sentences, text	Deep-FASP	Accuracy, MSE, R^2	–
[193]	Forums data	2004–2013	Sentences and keywords	Recursive neural tensor networks	Precision, recall, f-measure	–
[194]	News from Financial Times related US stocks	–	Sentiment of news headlines	SVR, Bidirectional LSTM	Cosine similarity	Python, Scikit Learn, Keras, Tensorflow

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:
Other text mining studies

Art.	Data set	Period	Feature set	Method	Performance criteria	**Env.**
[72]	News from NowNews, AppleDaily, LTN, MoneyDJ for 18 stocks	2013–2014	Text, Sentiment	DMLP	Return	Python, Tensorflow
[86]	The event data set for large European banks, news articles from Reuters	2007–2014	Word, sentence	DMLP +NLP preprocess	Relative usefulness, F1-score	–
[87]	Event dataset on European banks, news from Reuters	2007–2014	Text, sentence	Sentence vector + DFFN	Usefulness, F1-score, AUROC	–
[88]	News from Reuters, fundamental data	2007–2014	Financial ratios and news text	doc2vec + NN	Relative usefulness	Doc2vec
[121]	Real-world data for automobile insurance company labeled as fradulent	–	Car, insurance and accident related features	DMLP + LDA	TP, FP, Accuracy, Precision, F1-score	–
[123]	Financial transactions	–	Transaction data	LSTM	t-SNE	–
[195]	Taiwan's National Pension Insurance	2008–2014	Insured's id, area-code, gender, etc.	RNN	Accuracy, total error	Python
[196]	StockTwits	2015–2016	Sentences, StockTwits messages	Doc2vec, CNN	Accuracy, precision, recall, f-measure, AUC	Python, Tensorflow

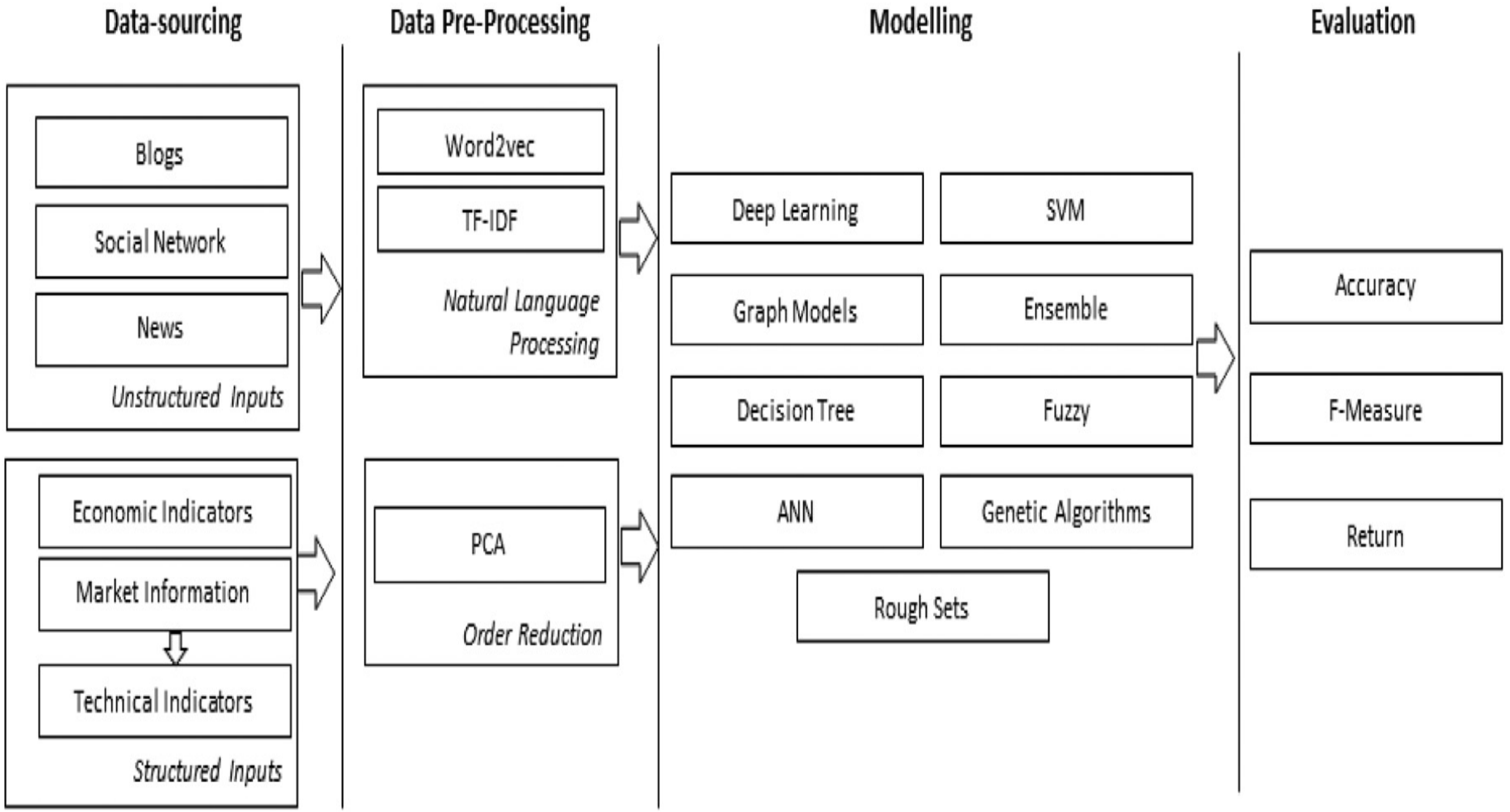
Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications:
Other financial applications

Art.	Subtopic	Data set	Period	Feature set	Method	Performance criteria	Env.
[47]	Improving trading decisions	S&P500, KOSPI, HSI, and EuroStoxx50	1987–2017	200-days stock price	Deep Q-Learning and DMLP	Total profit, Correlation	–
[193]	Identifying Top Sellers In Underground Economy	Forums data	2004–2013	Sentences and keywords	Recursive neural tensor networks	Precision, recall, f-measure	–
[195]	Predicting Social Ins. Payment Behavior	Taiwan's National Pension Insurance	2008–2014	Insured's id, area-code, gender, etc.	RNN	Accuracy, total error	Python
[199]	Speedup	45 CME listed commodity and FX futures	1991–2014	Price data	DNN	–	–
[200]	Forecasting Fundamentals	Stocks in NYSE, NASDAQ or AMEX exchanges	1970–2017	16 fundamental features from balance sheet	DMLP, LFM	MSE, Compound annual return, SR	–
[201]	Predicting Bank Telemarketing	Phone calls of bank marketing data	2008–2010	16 finance-related attributes	CNN	Accuracy	–
[202]	Corporate Performance Prediction	22 pharmaceutical companies data in US stock market	2000–2015	11 financial and 4 patent indicator	RBM, DBN	RMSE, profit	–

Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Stock Market Movement Forecast:
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.

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.

Deep Learning
and
Neural Networks

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

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

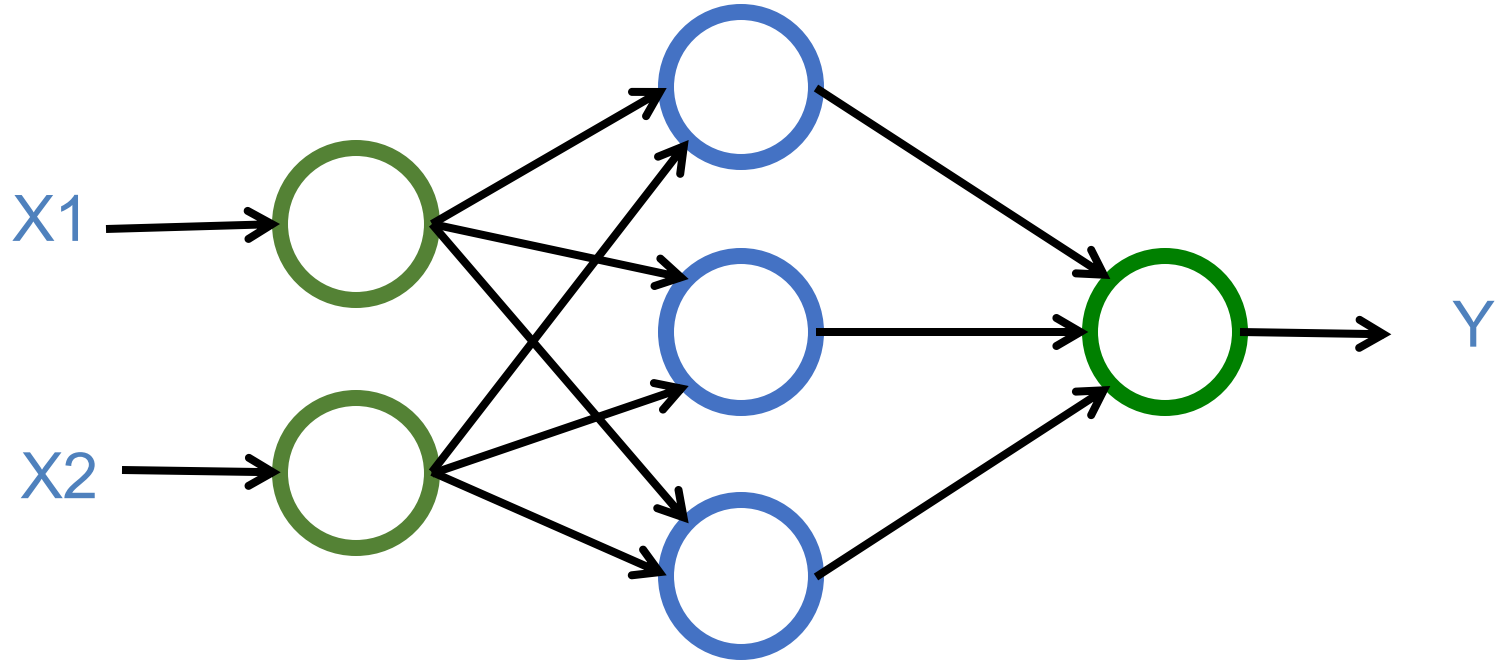
Tensor

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

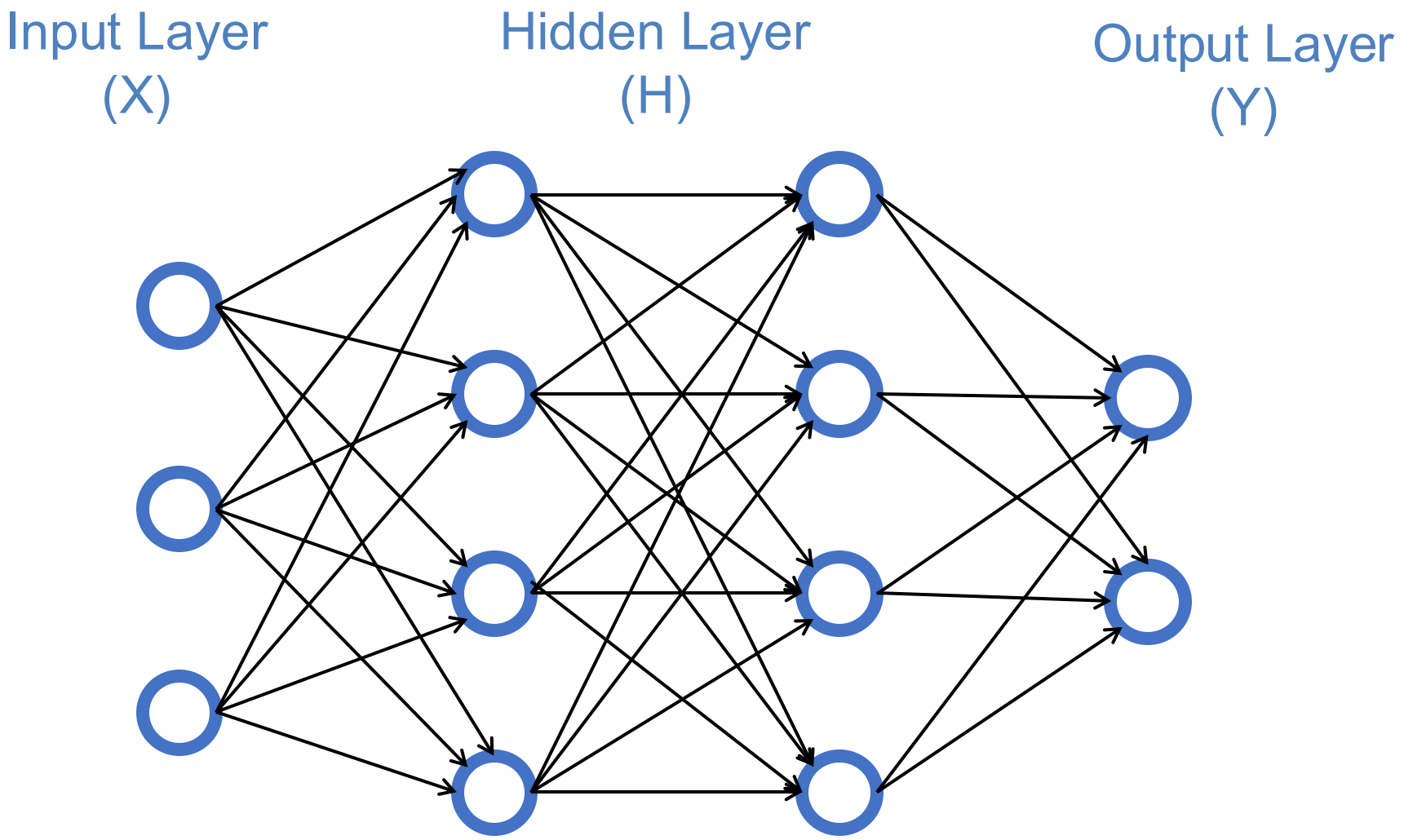
Deep Learning Foundations:
Neural Networks

Deep Learning and Neural Networks

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



Deep Learning and Neural Networks



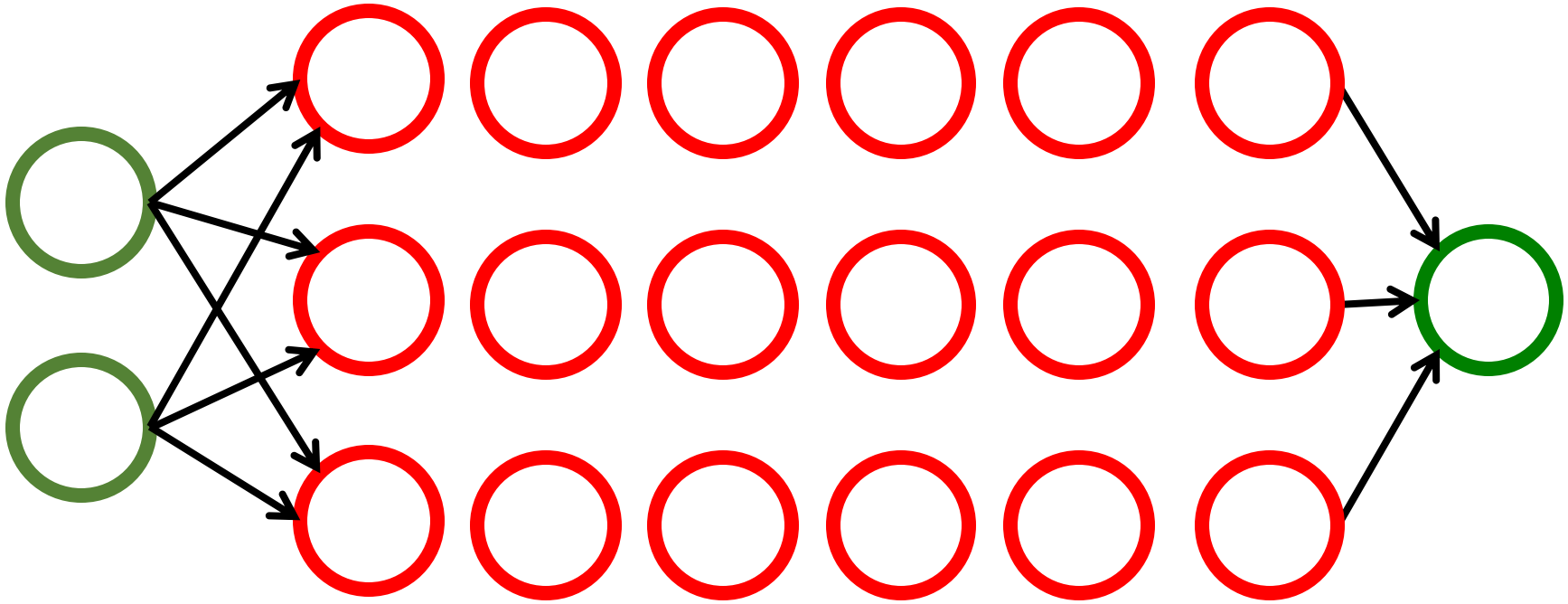
Deep Learning and Neural Networks

Input Layer (X)

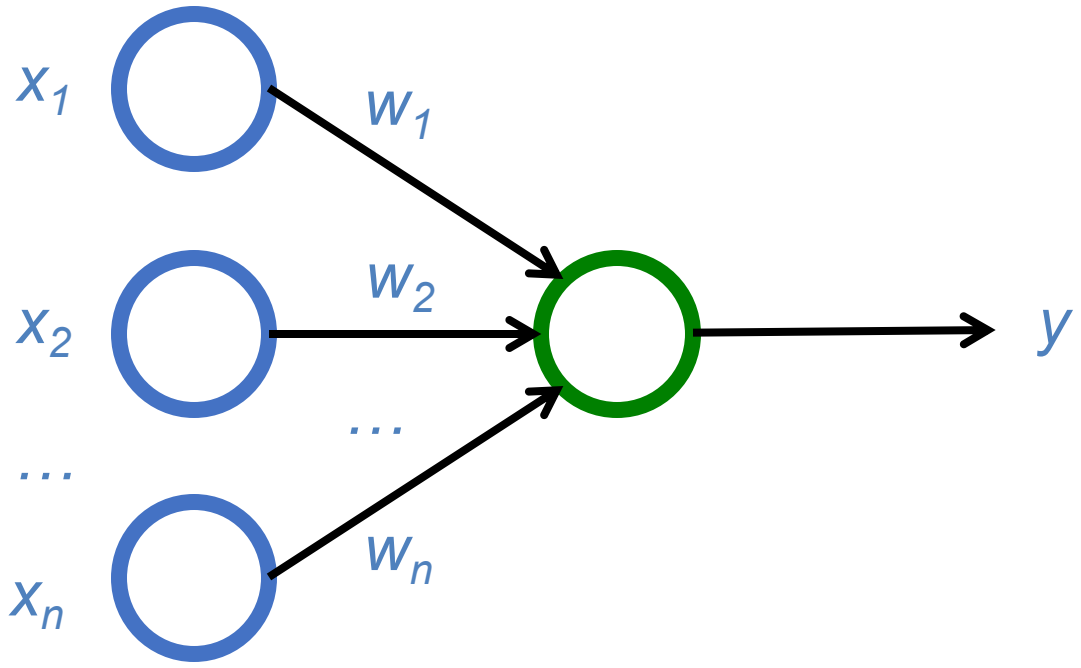
Hidden Layers (H)

Output Layer (Y)

Deep Neural Networks
Deep Learning



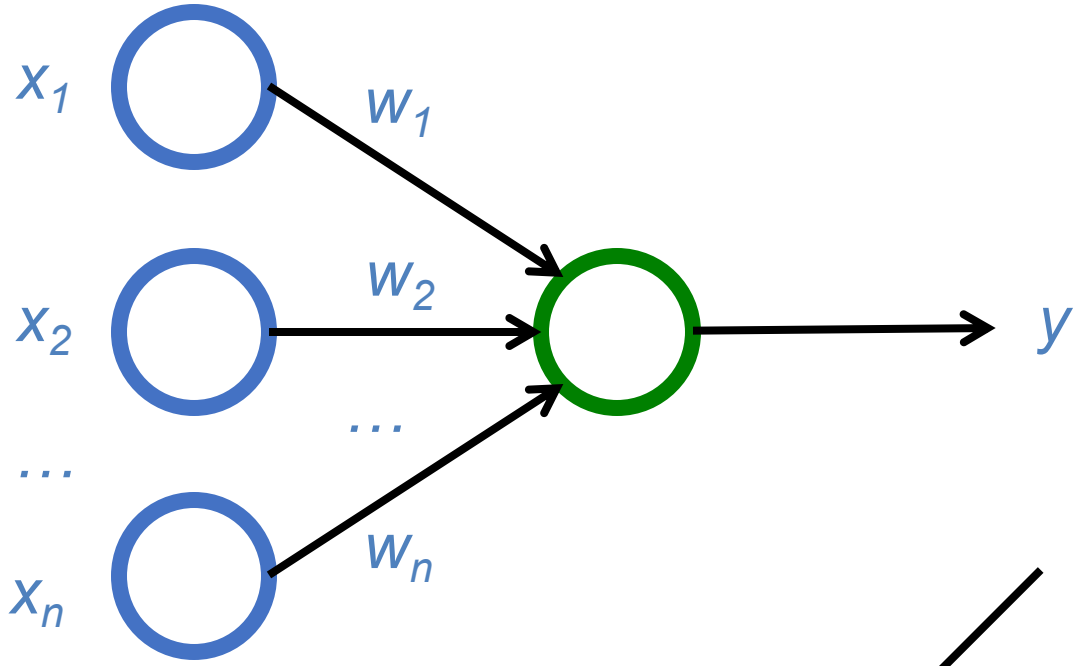
The Neuron

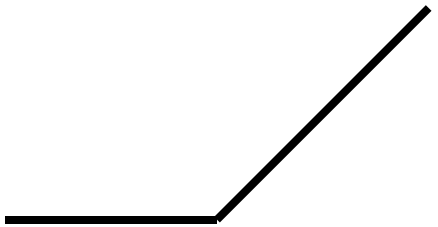


Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

The Neuron

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

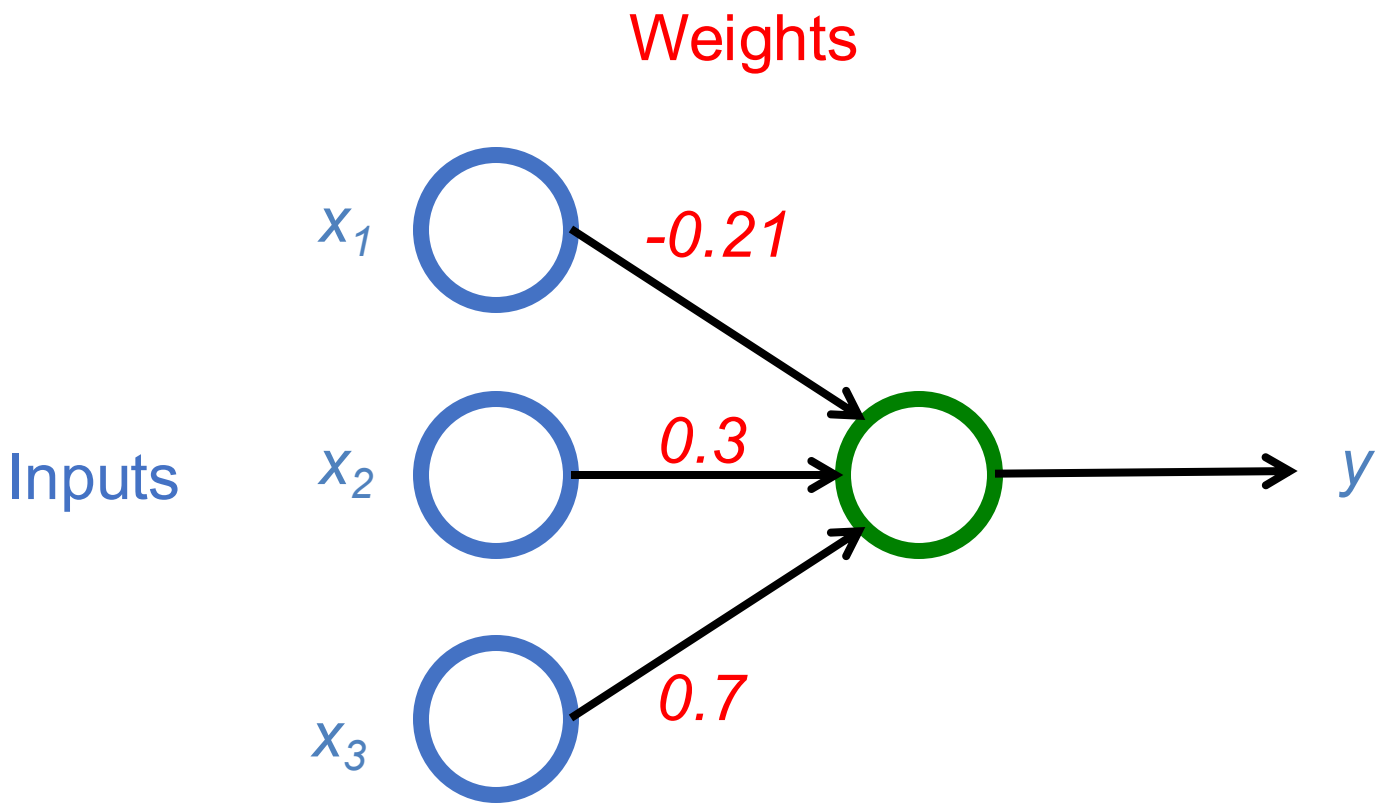




$$F(x) = \max(0, x)$$

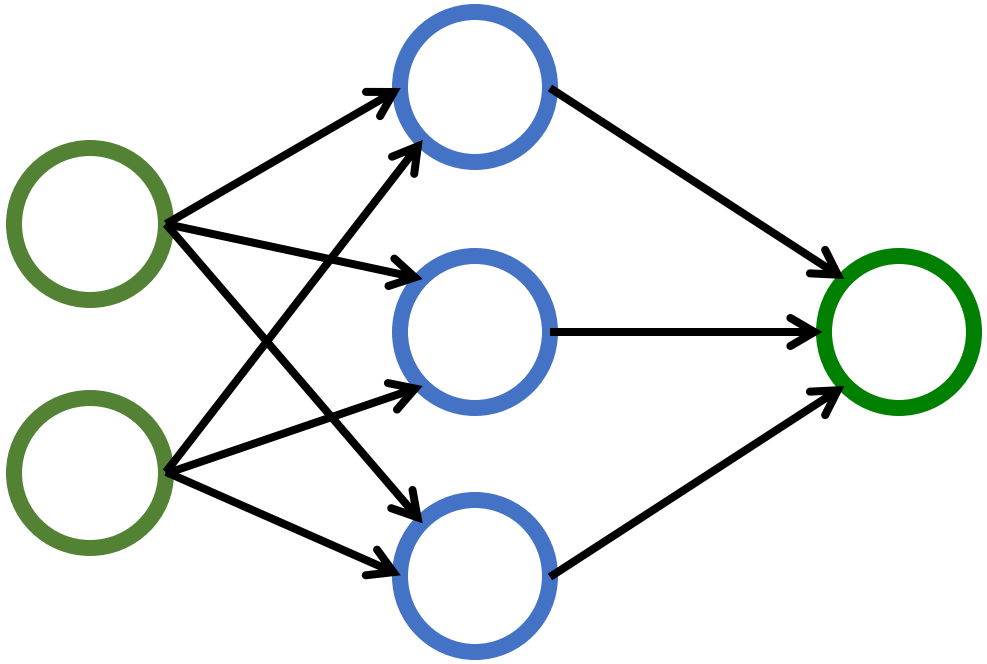
Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

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



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

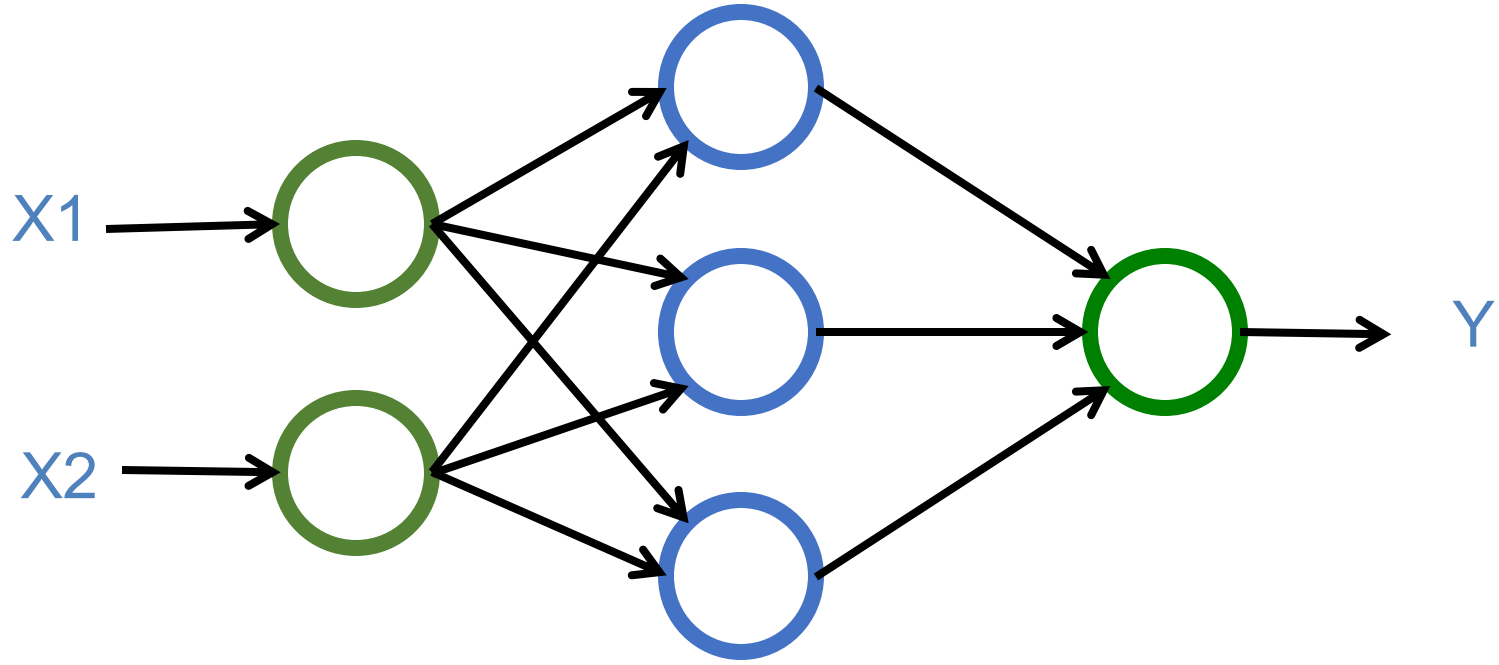
Neural Networks



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

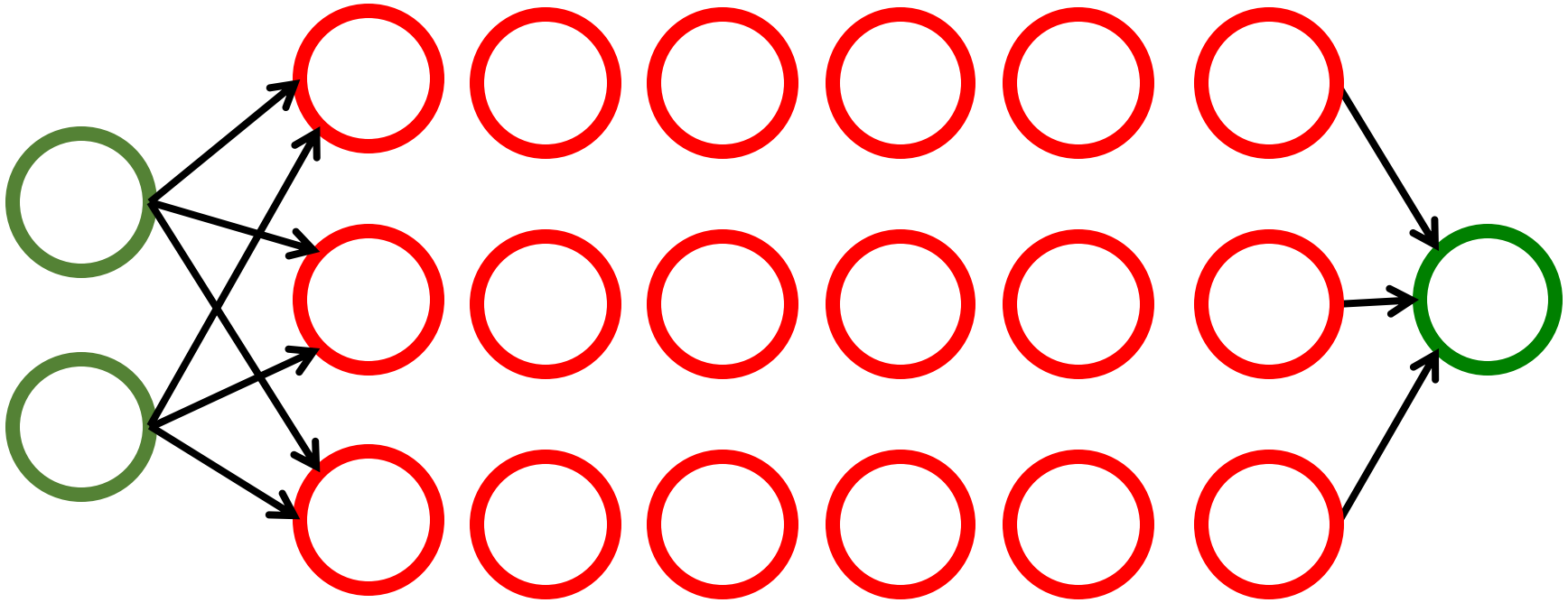
Neural Networks

Input Layer
(X)

Hidden Layers
(H)

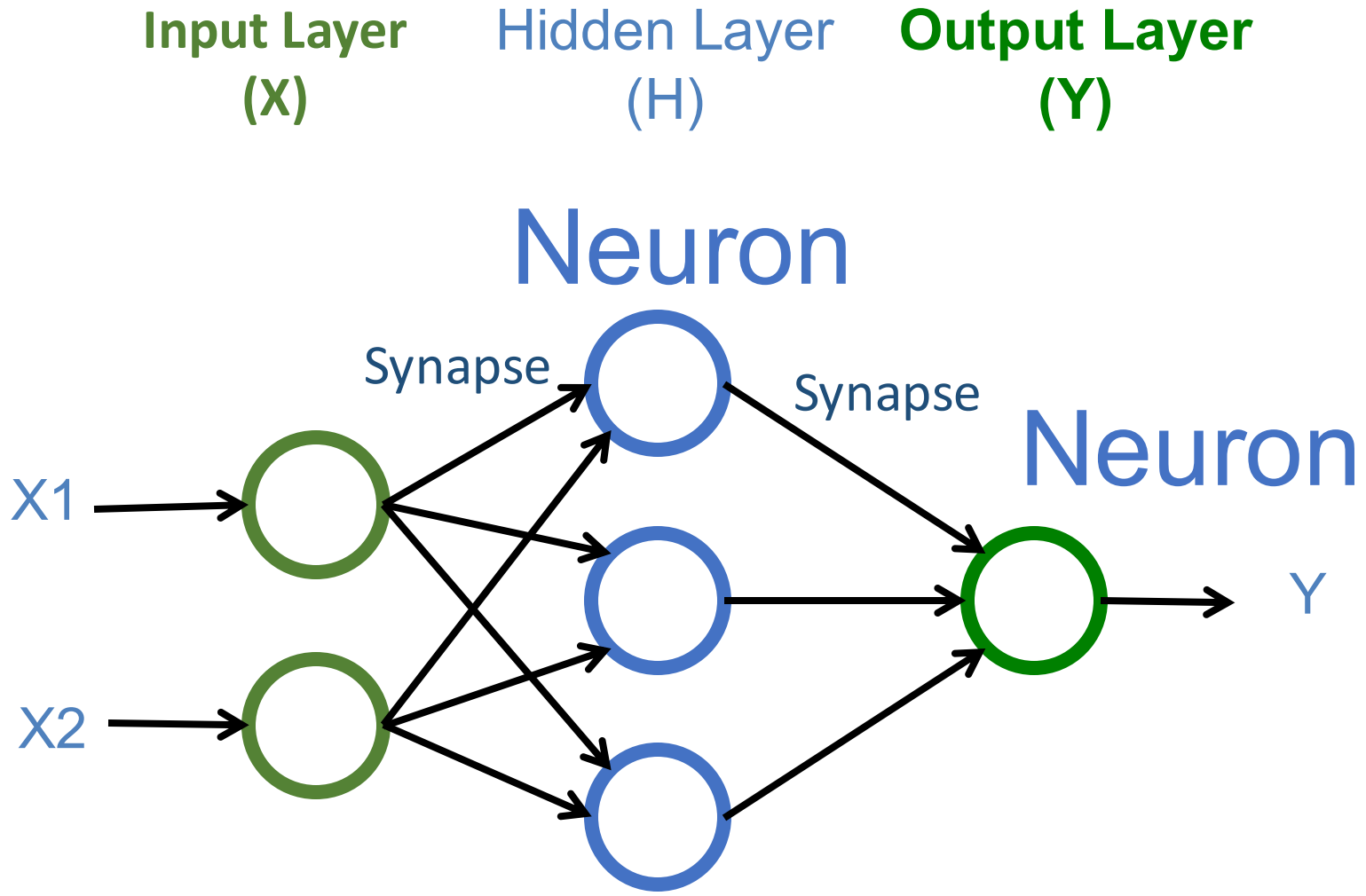
Output Layer
(Y)

Deep Neural Networks
Deep Learning



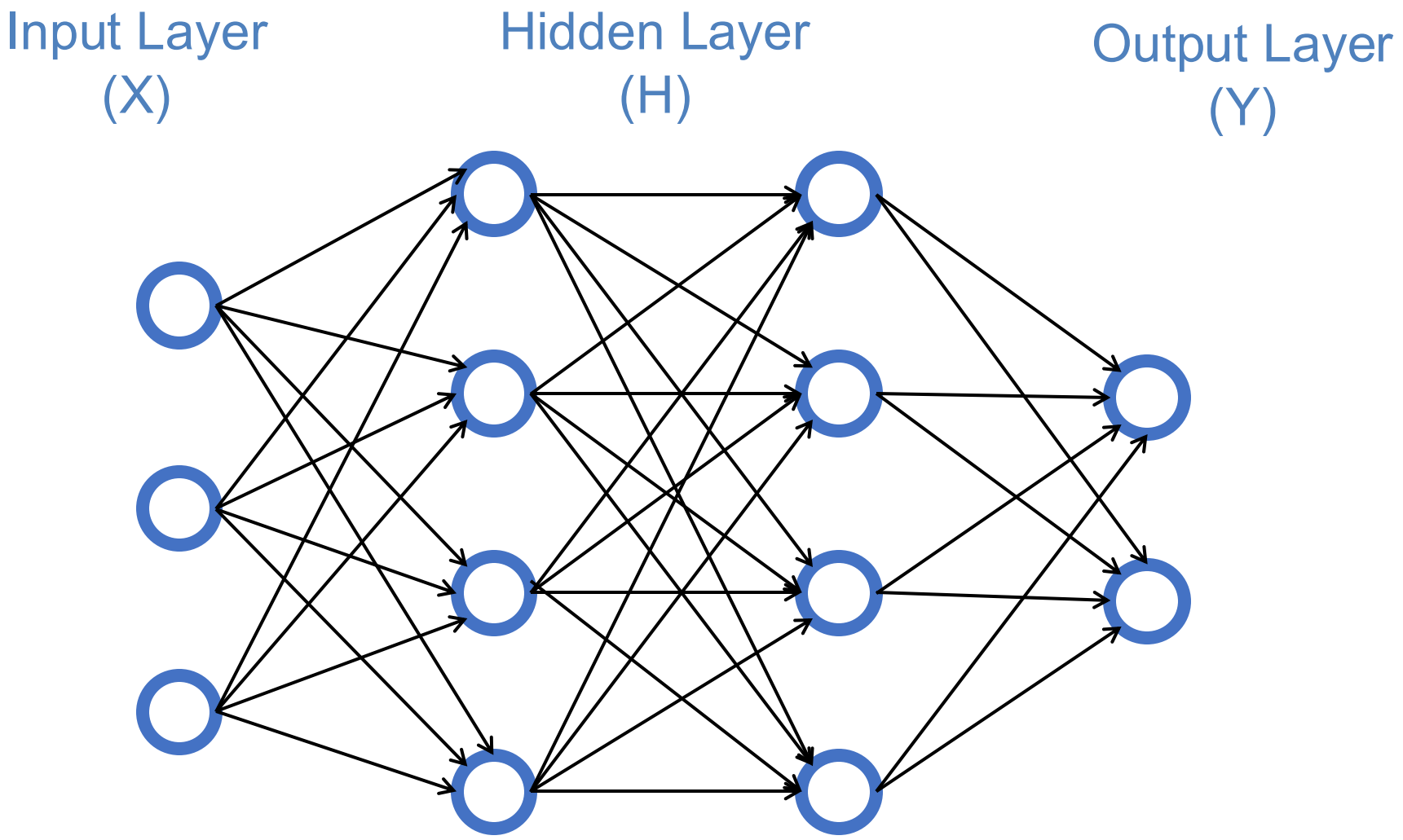
Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Neural Networks



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

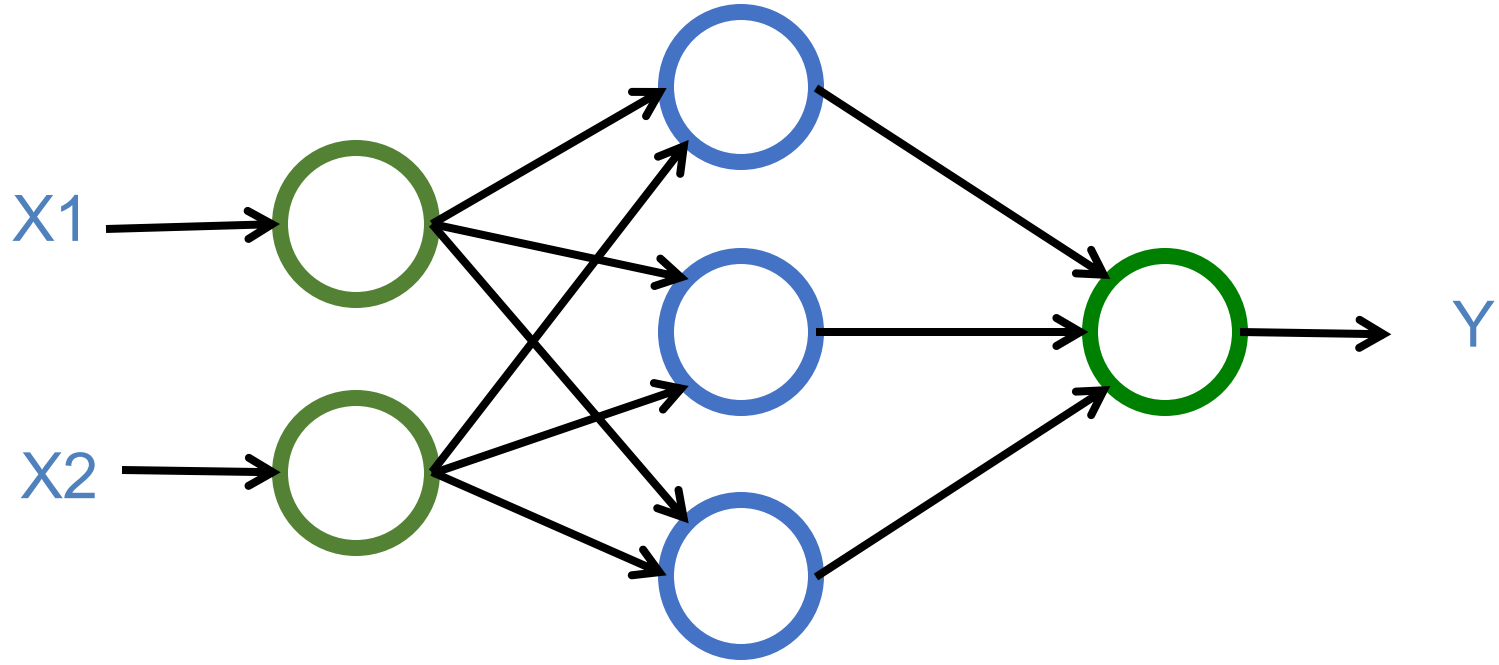
Neural Networks



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

Linear function

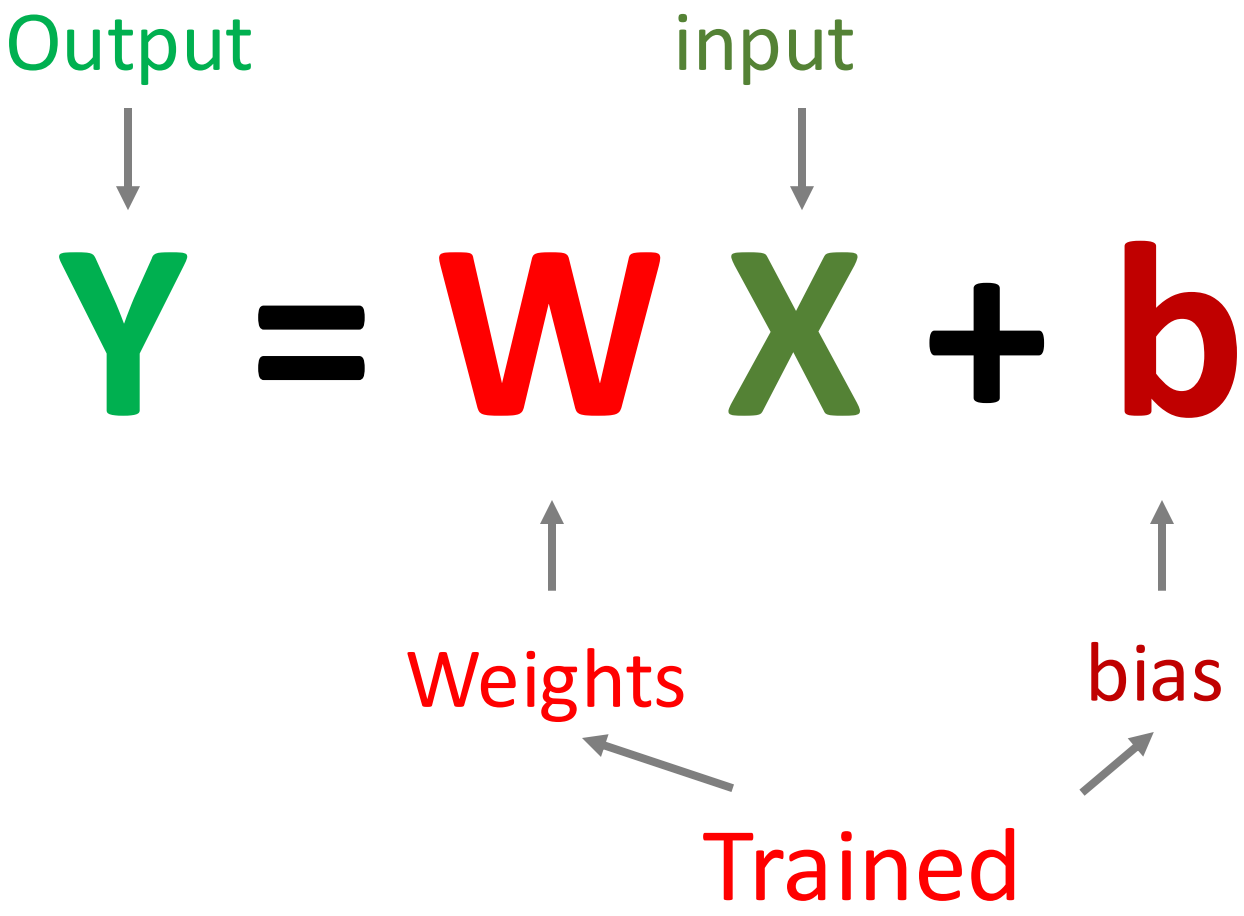
$$y = f(x)$$

$$y = w_1 x + w_0$$

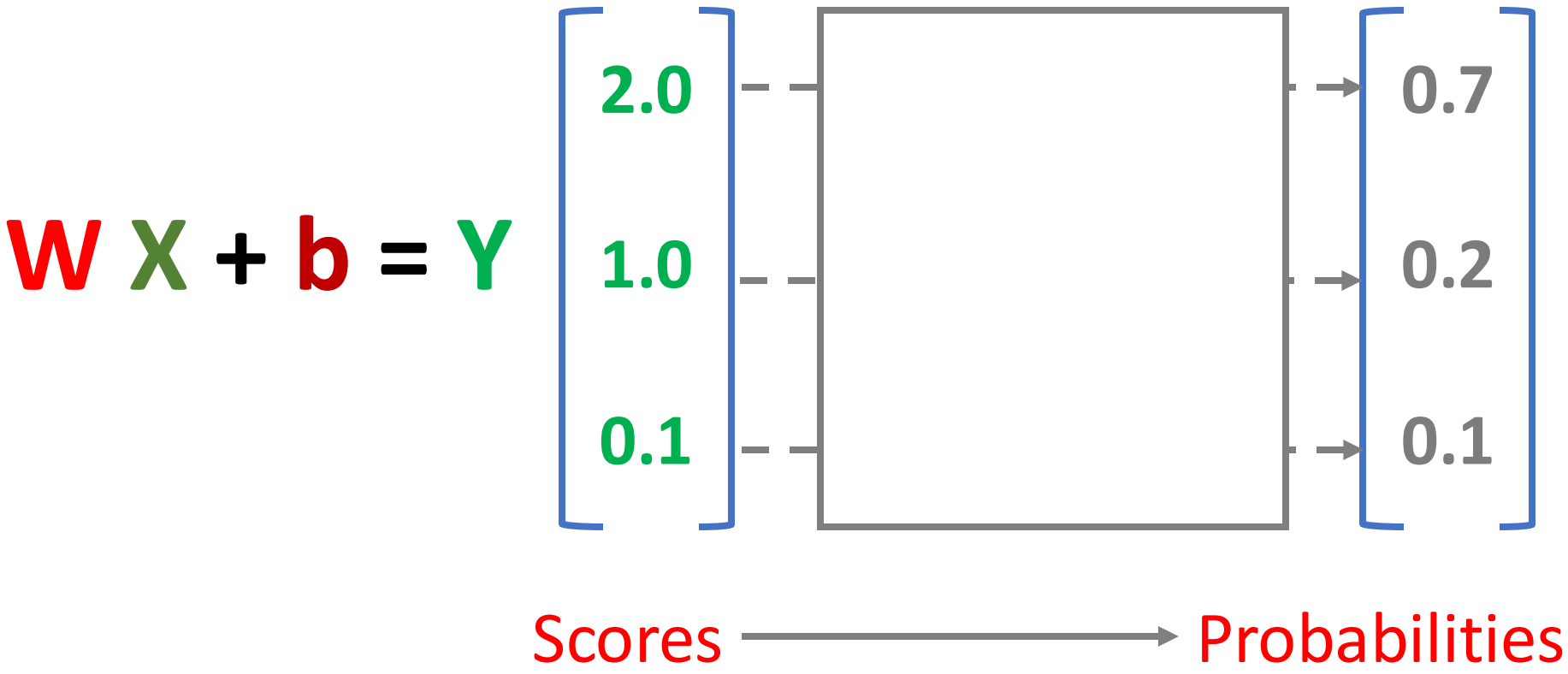
$$h_w(x) = w_1 x + w_0$$

$$Y = W X + b$$

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



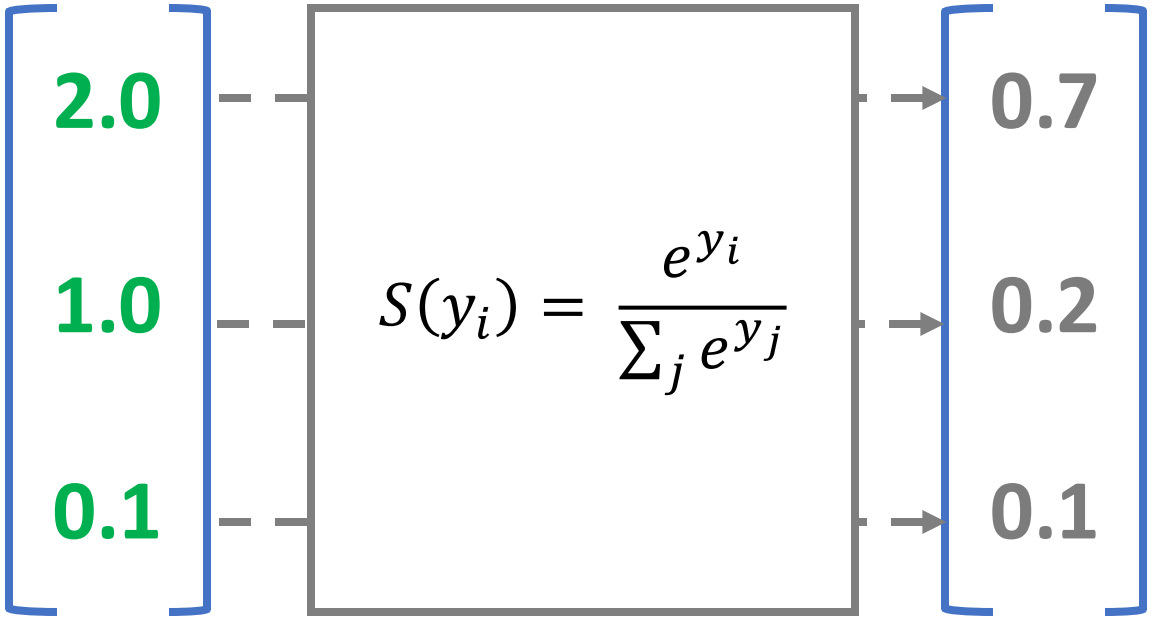
Source: https://www.youtube.com/watch?v=G8eNWzxOgqE



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

SoftMAX

$$W\ X + b = Y$$



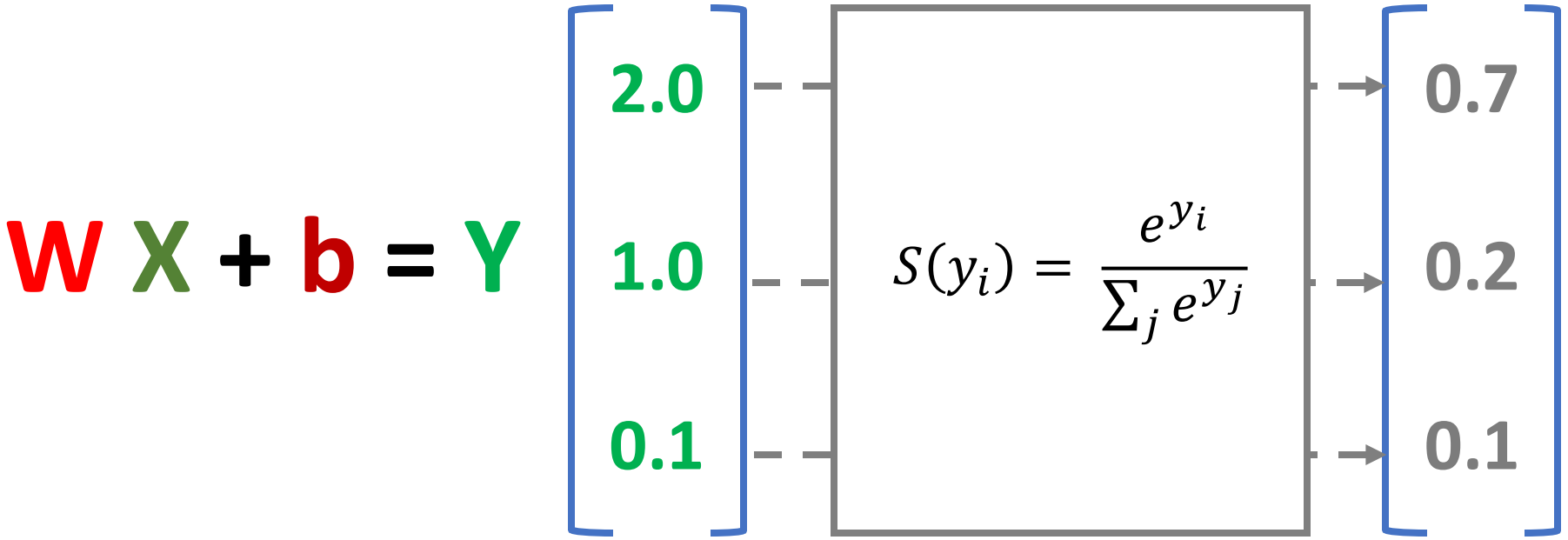
Logits Scores → Probabilities

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

=

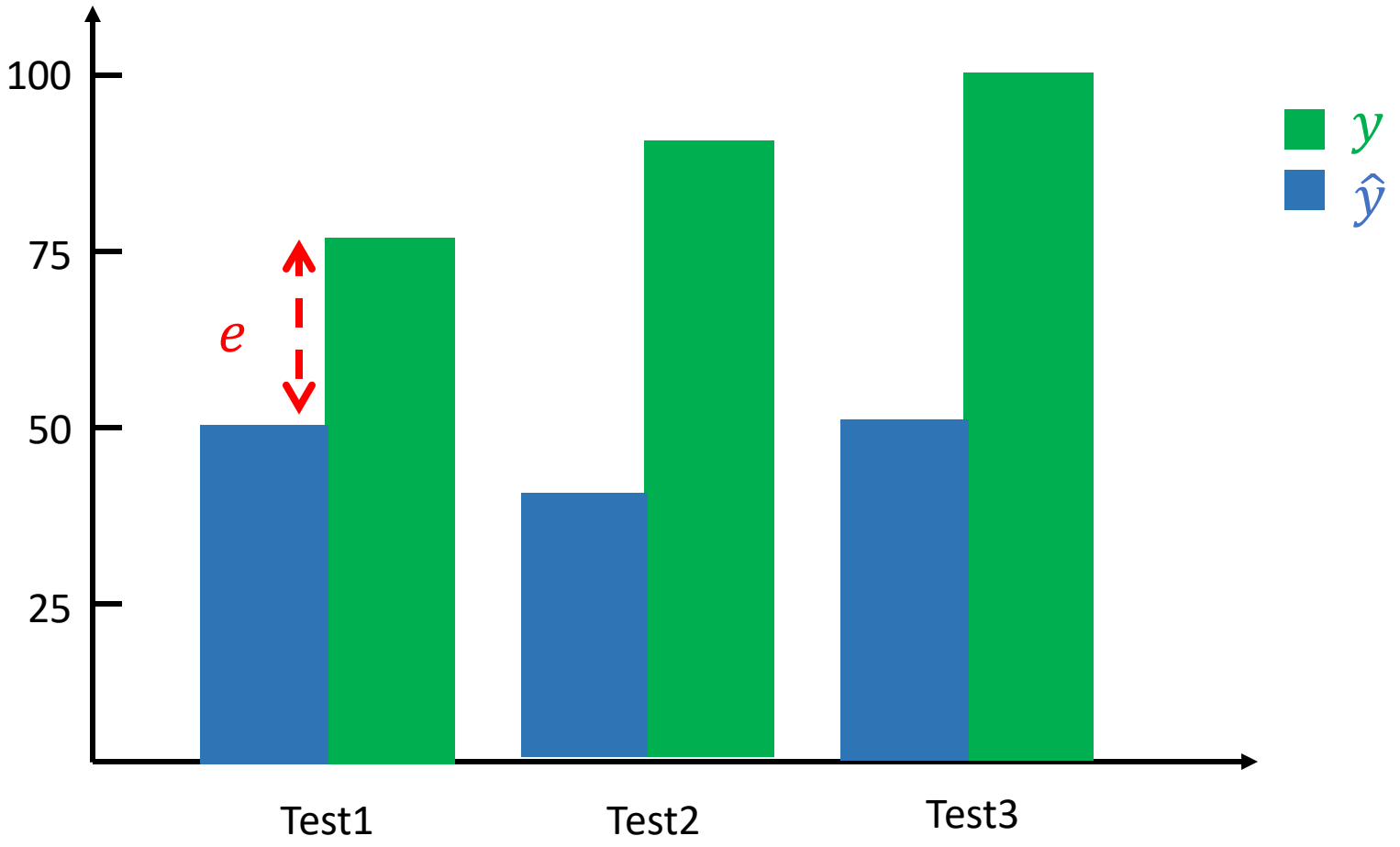
Minimize the Cost Function

Minimize the Loss Function

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

Error = Predict Y - Actual Y

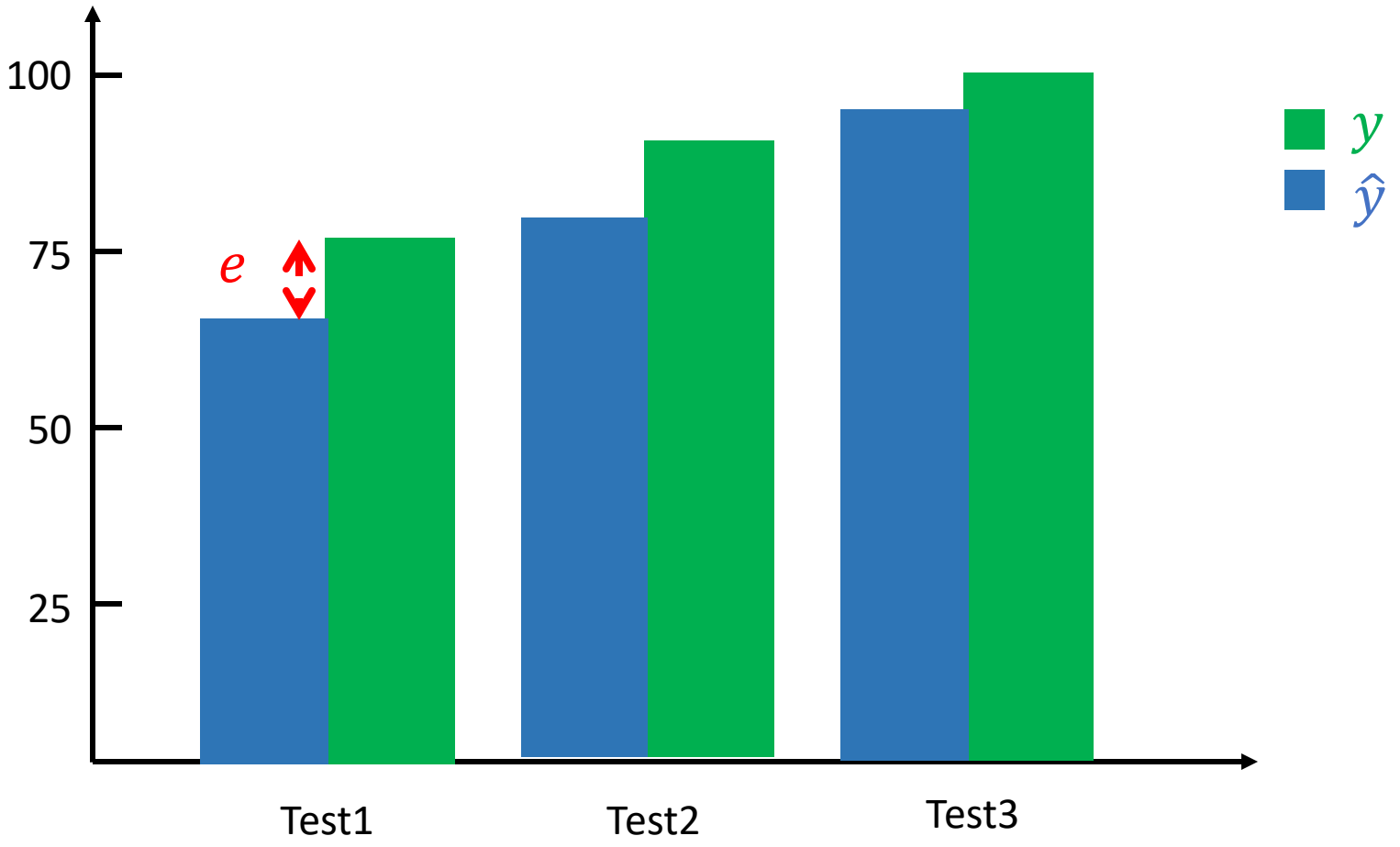
Error : Cost : Loss



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

Error = Predict Y - Actual Y

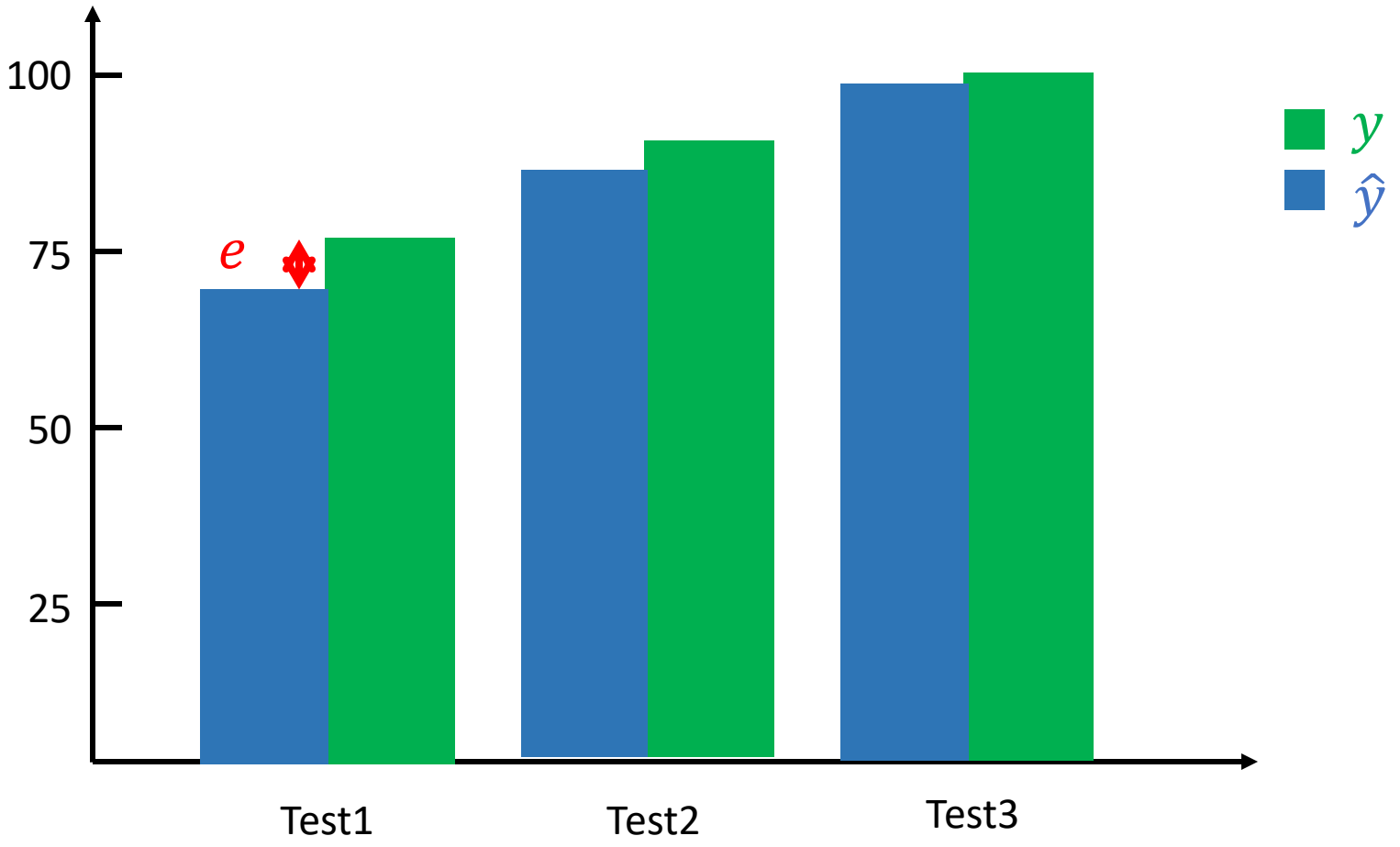
Error : Cost : Loss



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

Error = Predict Y - Actual Y

Error : Cost : Loss

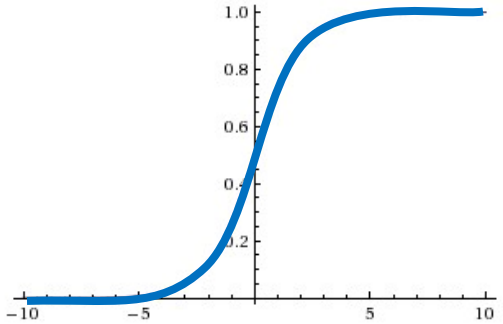


Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Activation Functions

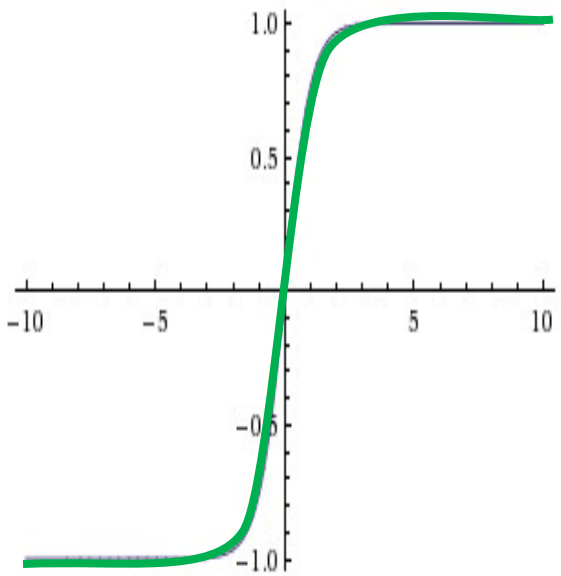
Activation Functions

Sigmoid



[0, 1]

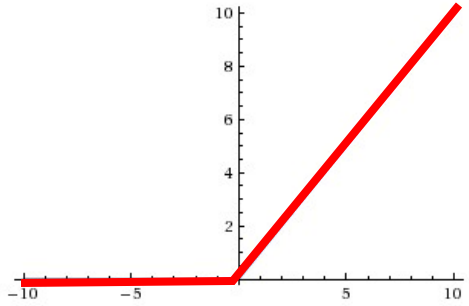
TanH



[-1, 1]

ReLU

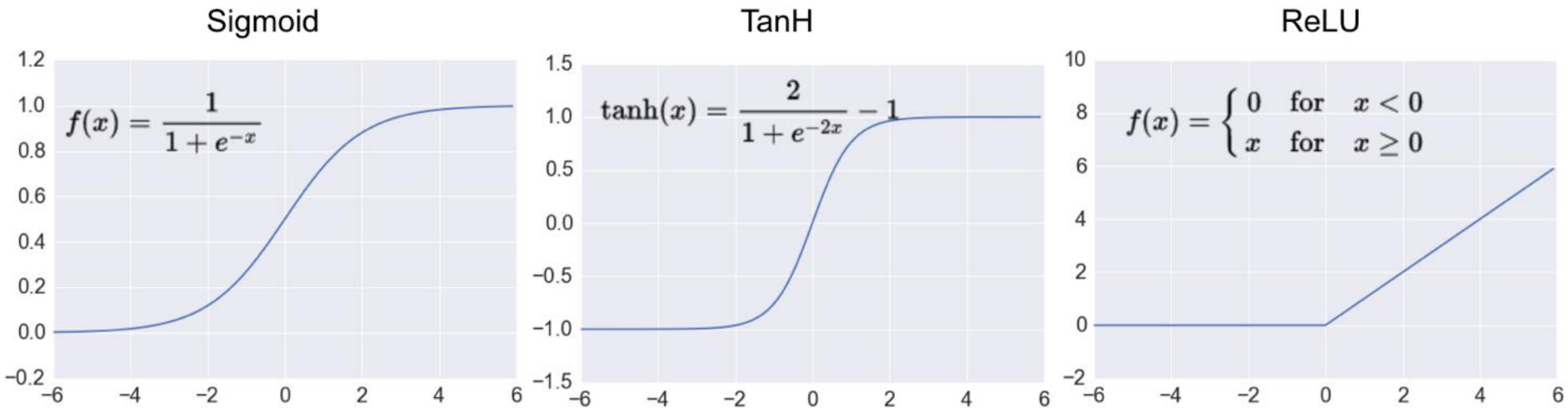
(Rectified Linear Unit)



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

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

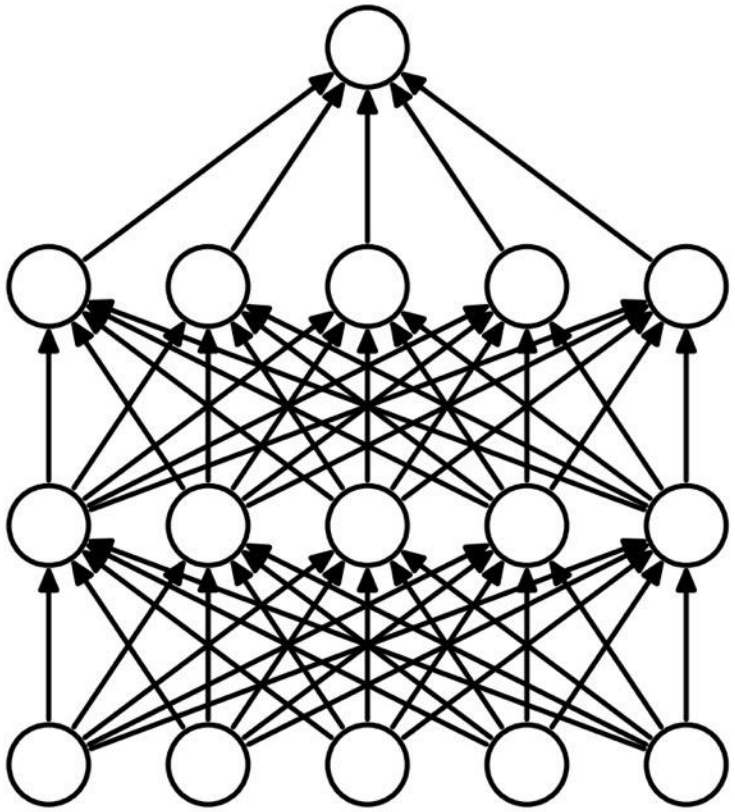
Multiple Classification: 10 Class

Activation Function:
SoftMAX

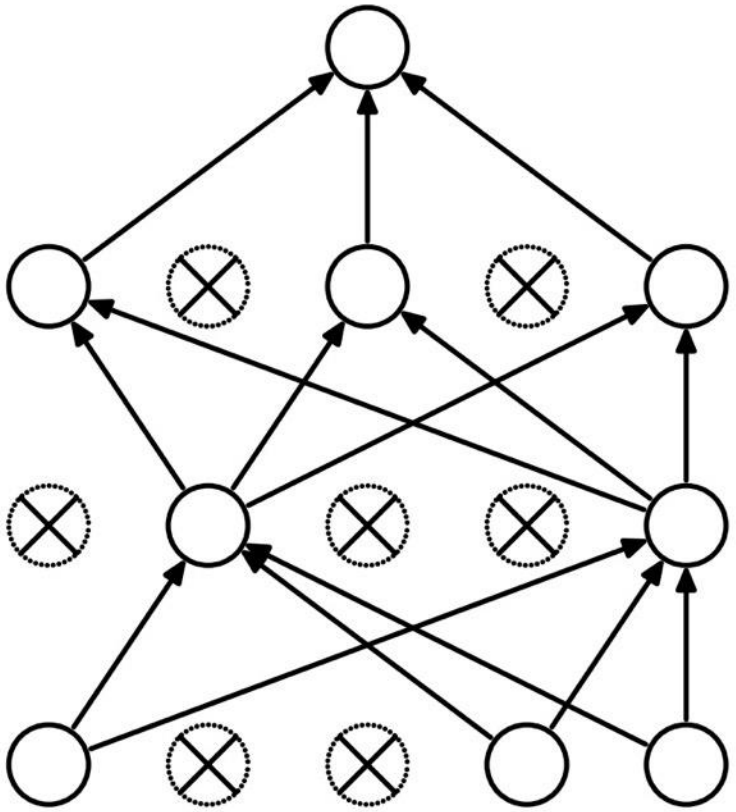
Loss Function:
Categorical Cross-Entropy

Dropout

Dropout: a simple way to prevent neural networks from overfitting



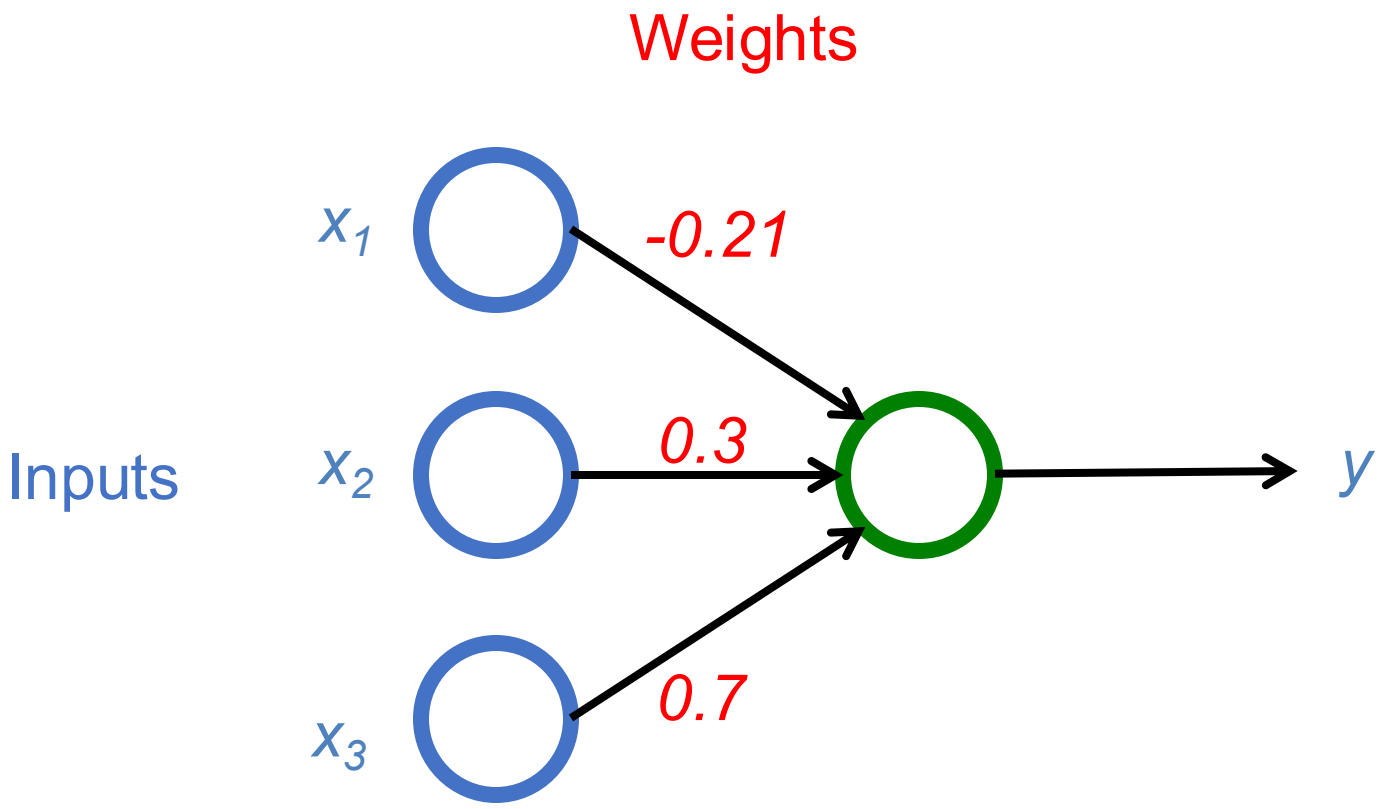
(a) Standard Neural Net



(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.

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



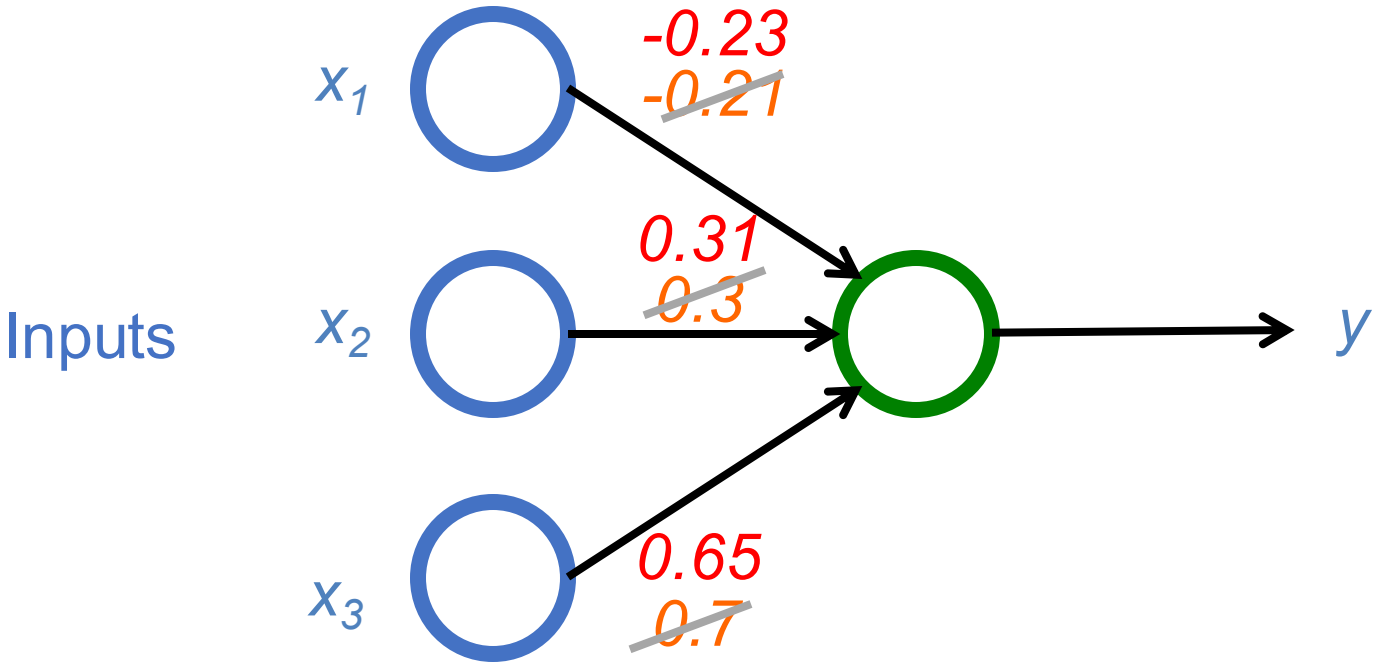
Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

Next time:

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

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

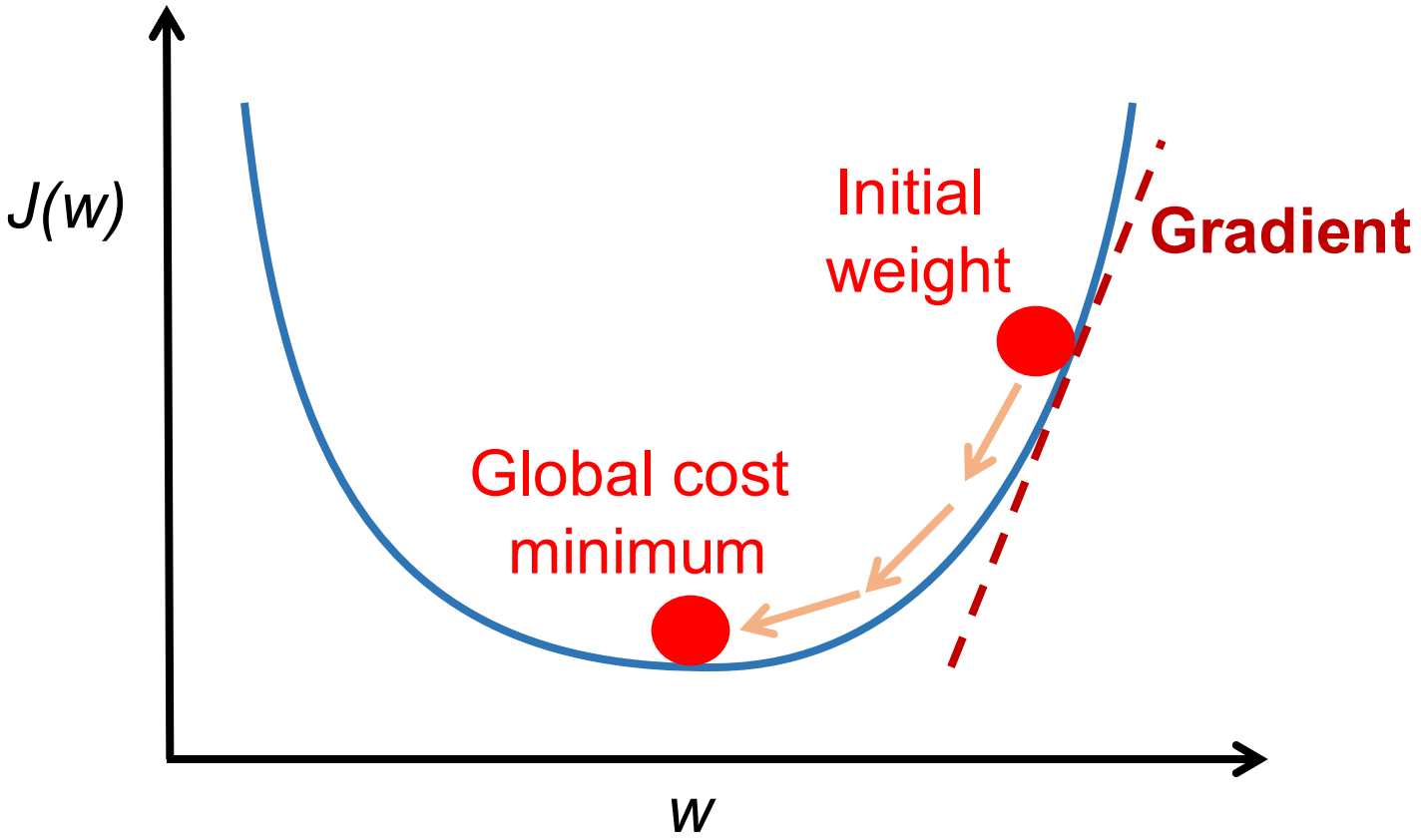
Weights

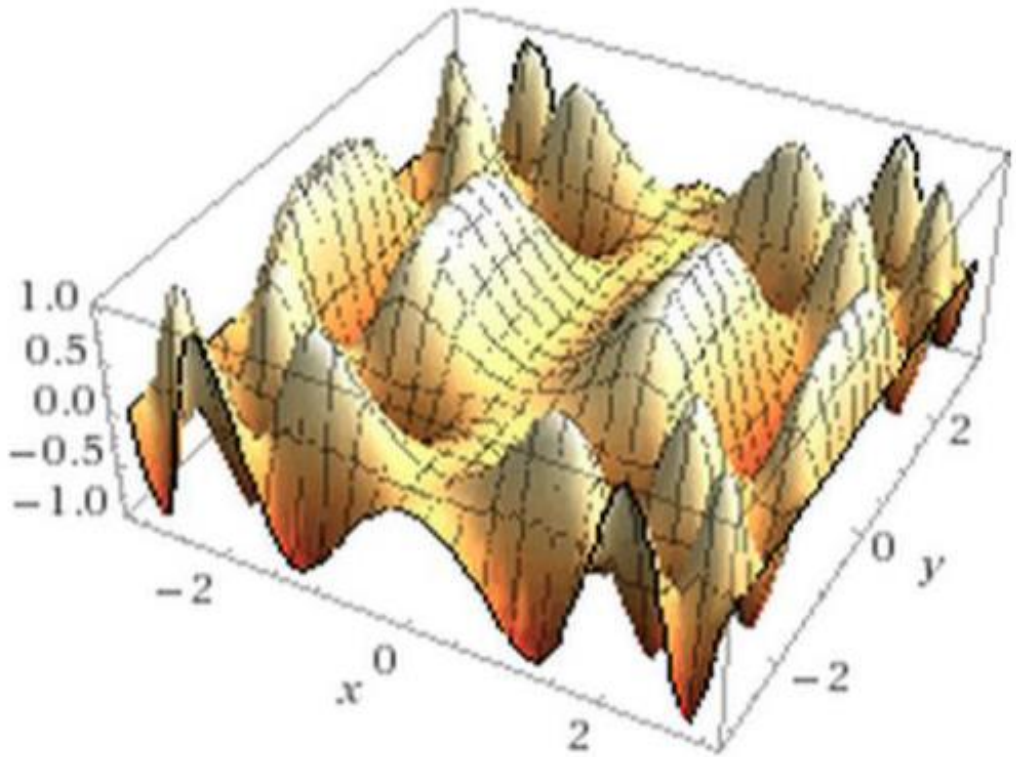


Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

Optimizer:

Stochastic Gradient Descent (SGD)





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

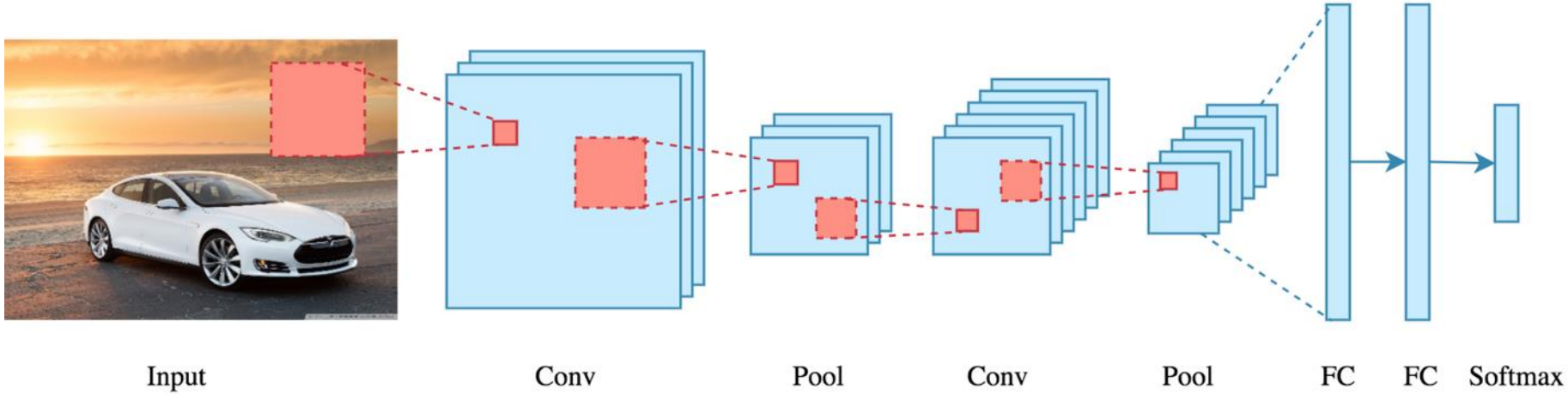
Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

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

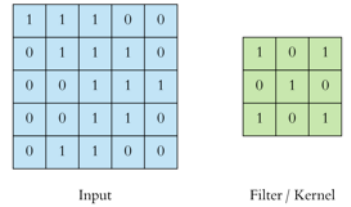
Input x Filter

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		

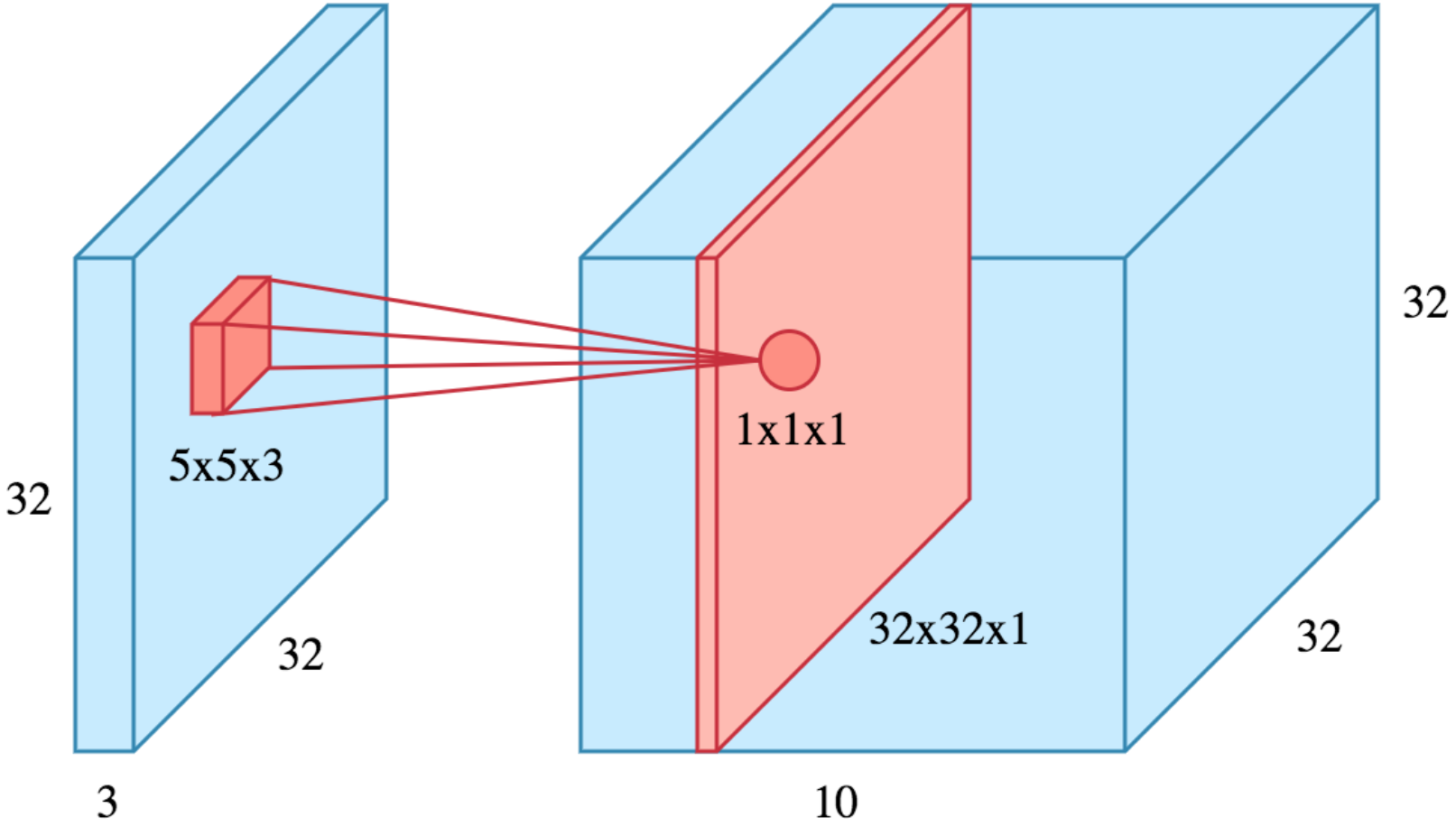
Example convolution operation shown in 2D using a 3x3 filter

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

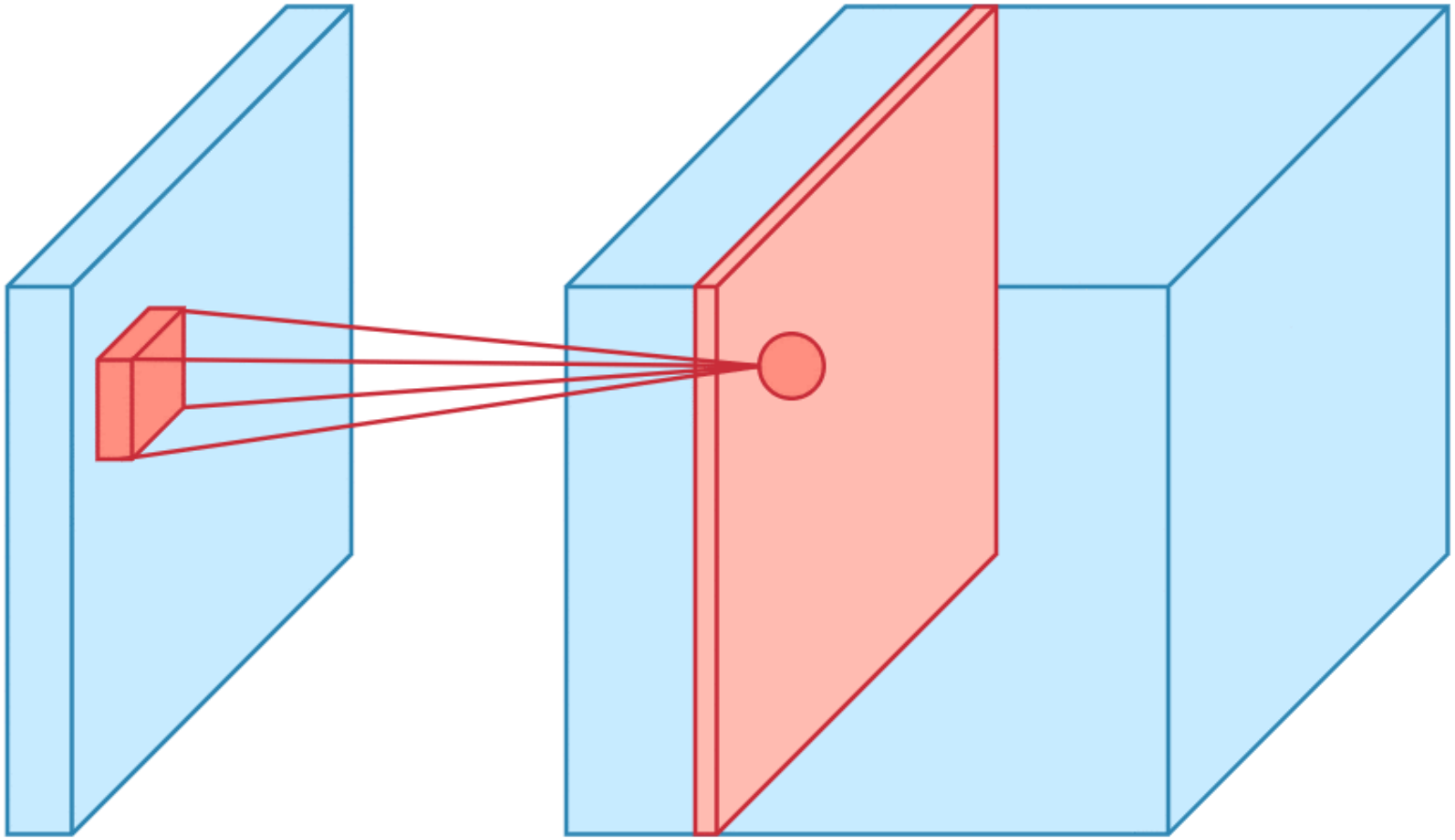


final output of the convolution layer:
a volume of size 32x32x10

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

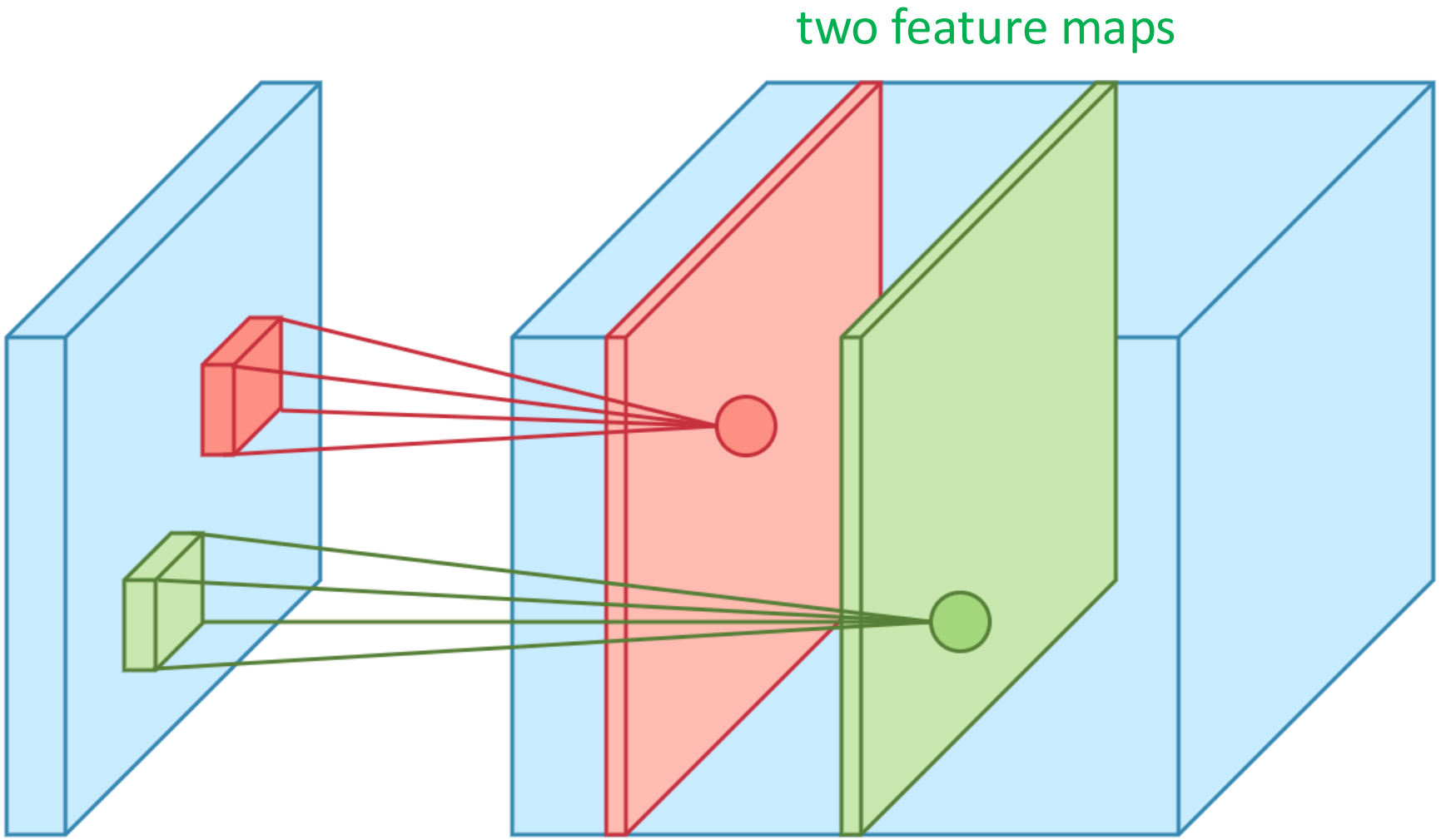
CNN Convolution Layer

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

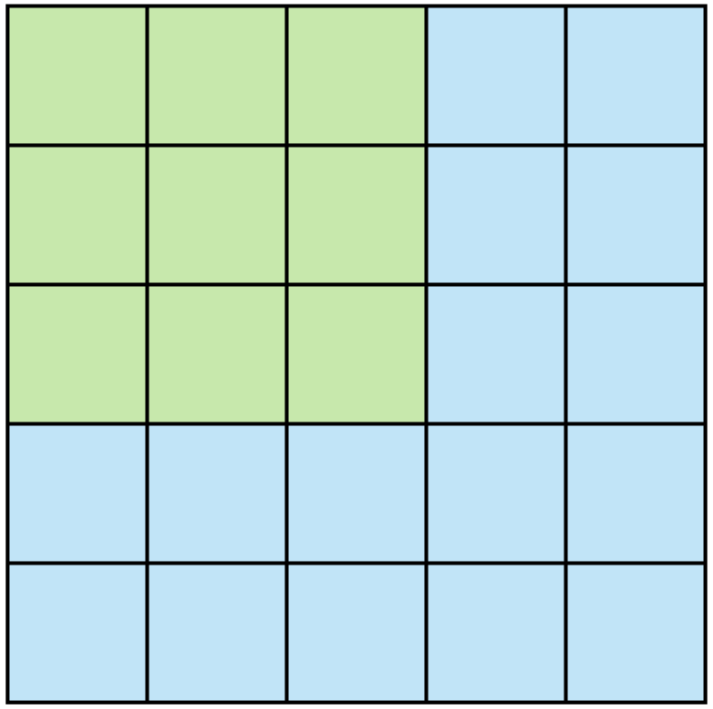
CNN Convolution Layer



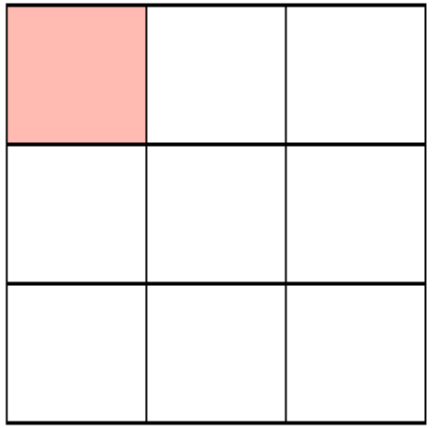
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks,
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Convolution Layer

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



Stride 1

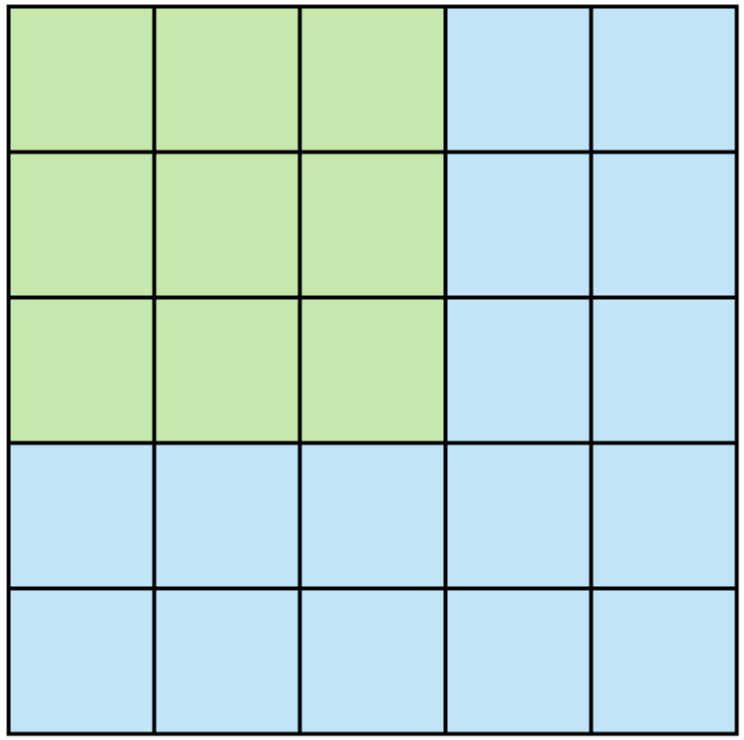


Feature Map

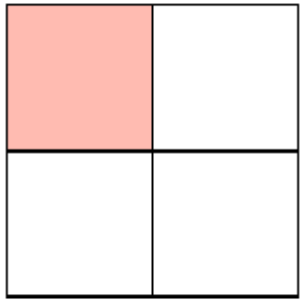
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks,
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Convolution Layer

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



Stride 2

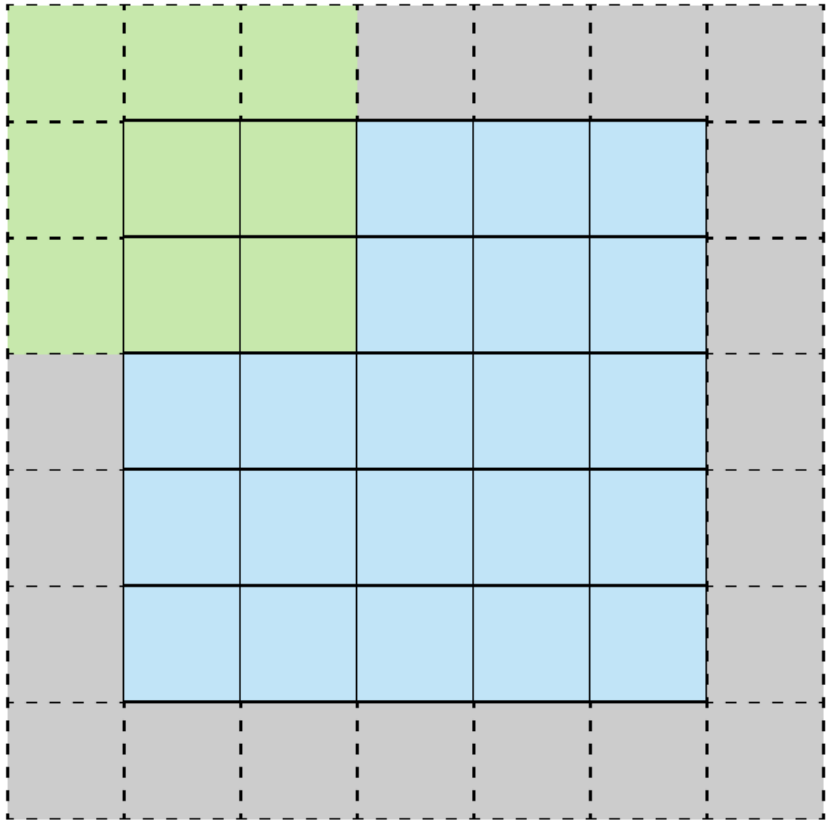


Feature Map

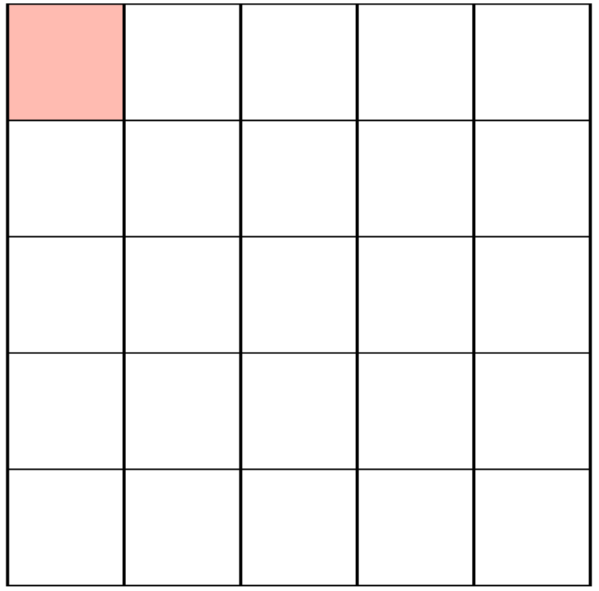
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks,
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

CNN Convolution Layer

Stride 1 with Padding



Stride 1 with Padding

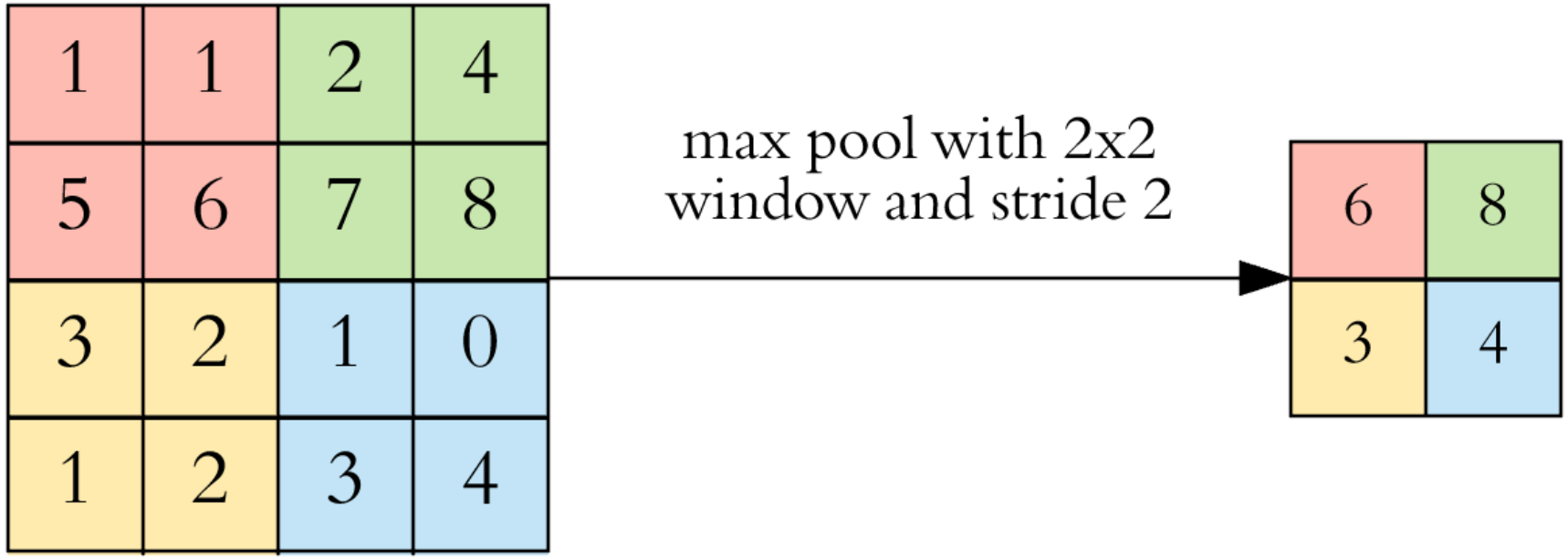


Feature Map

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

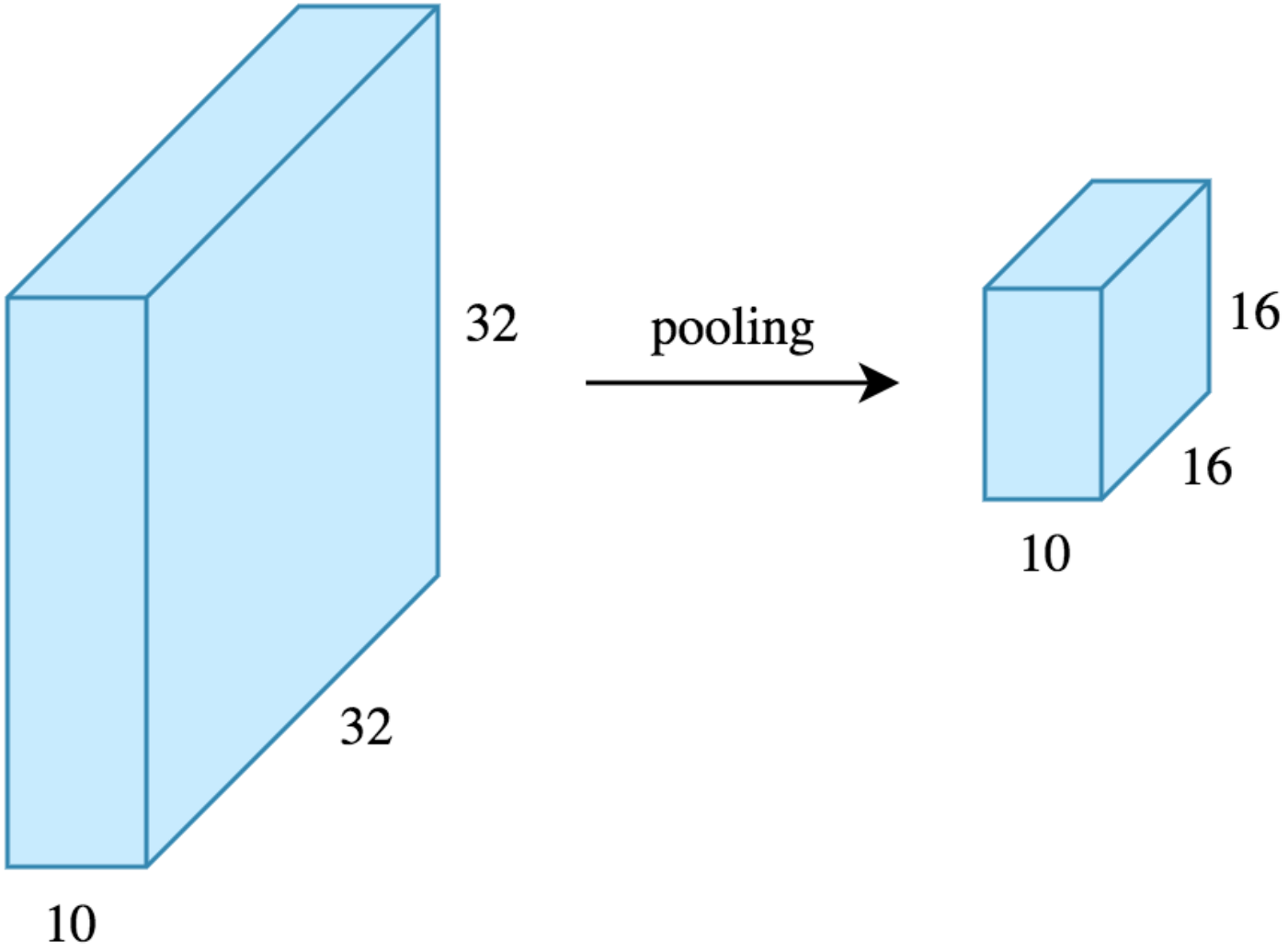
CNN Pooling Layer

Max Pooling



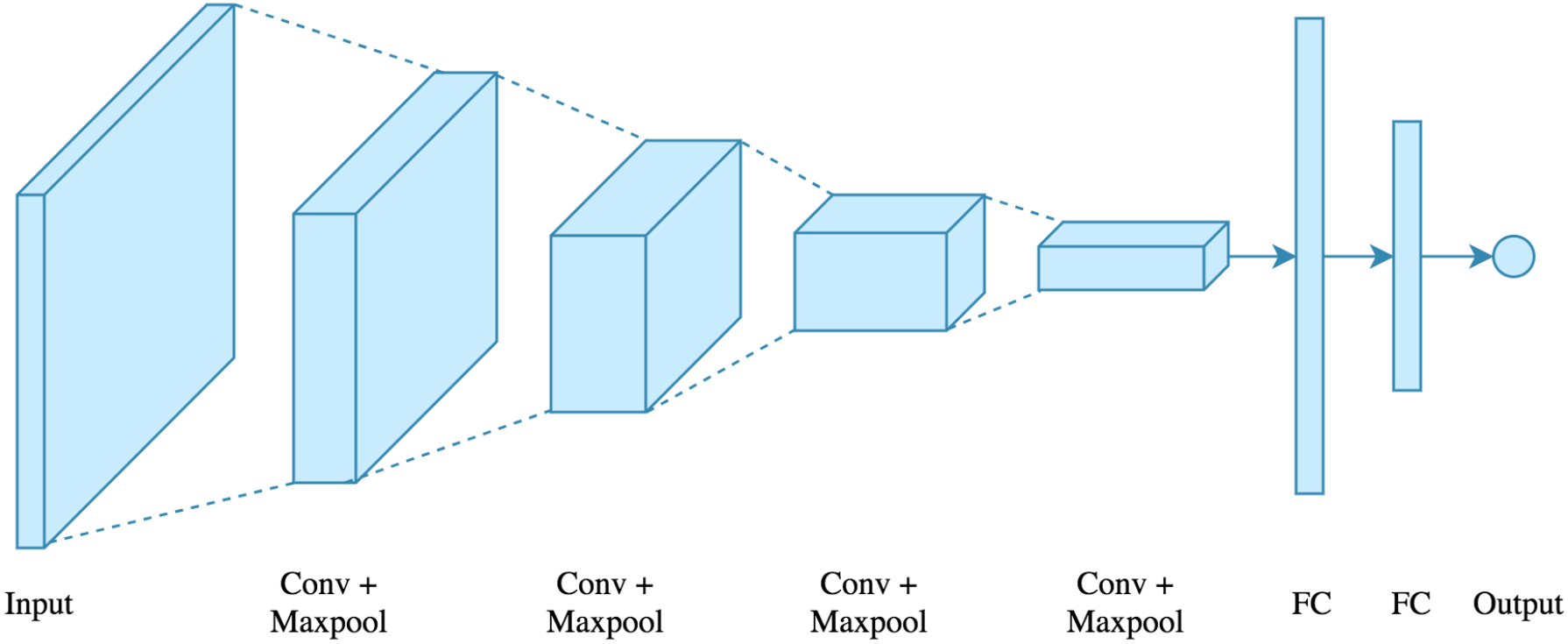
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks,
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

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

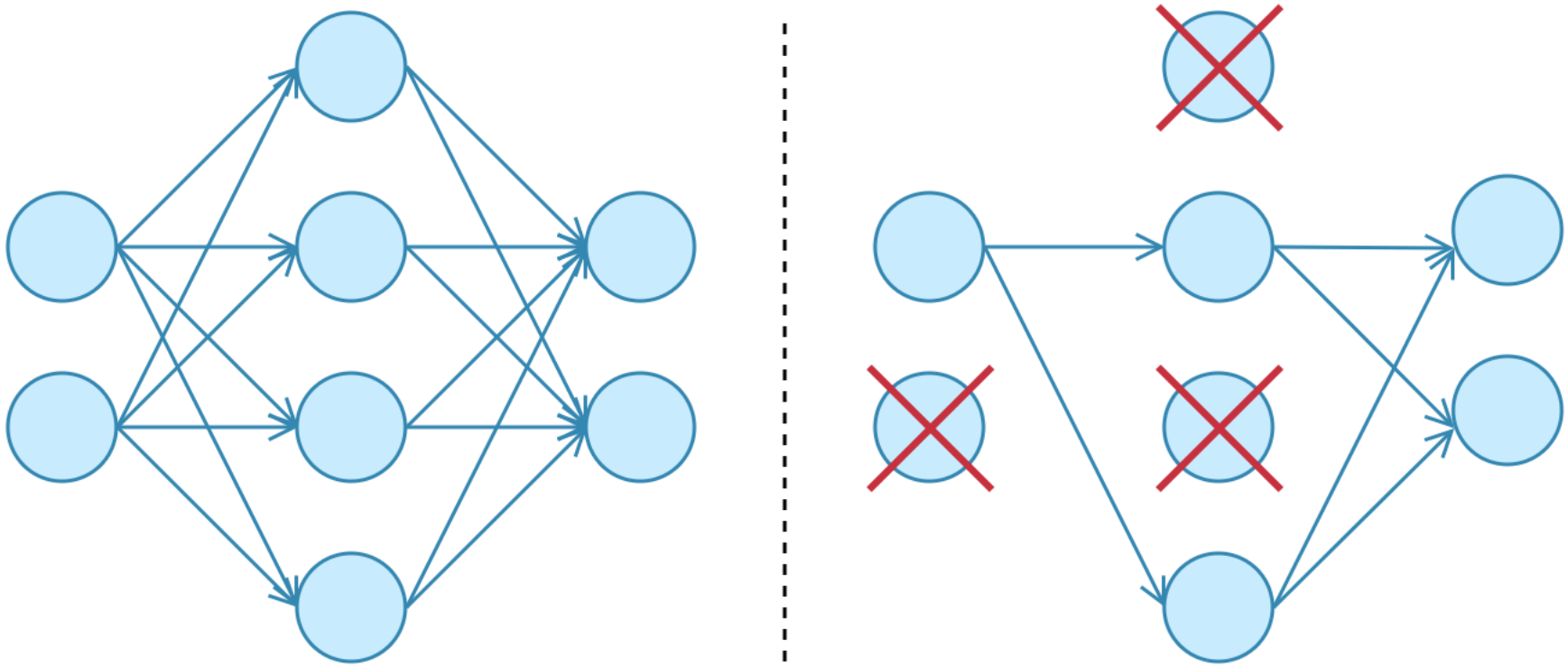
CNN Architecture
4 convolution + pooling layers, followed by 2 fully connected layers

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

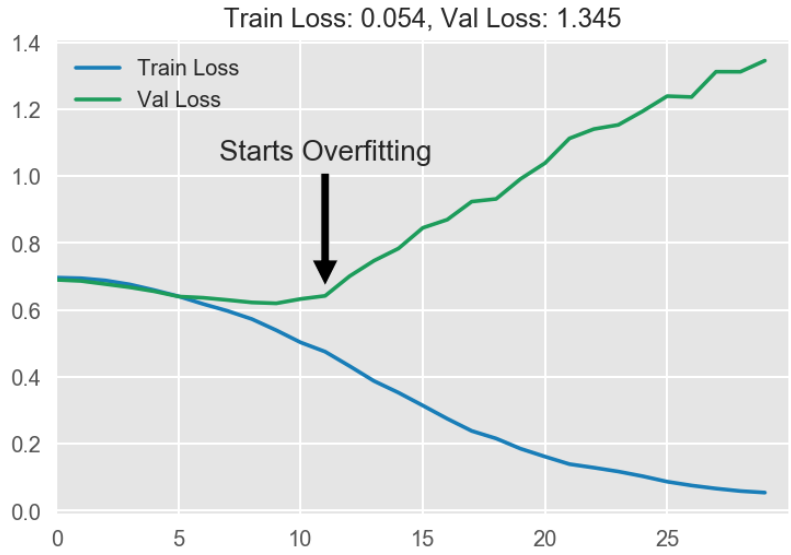


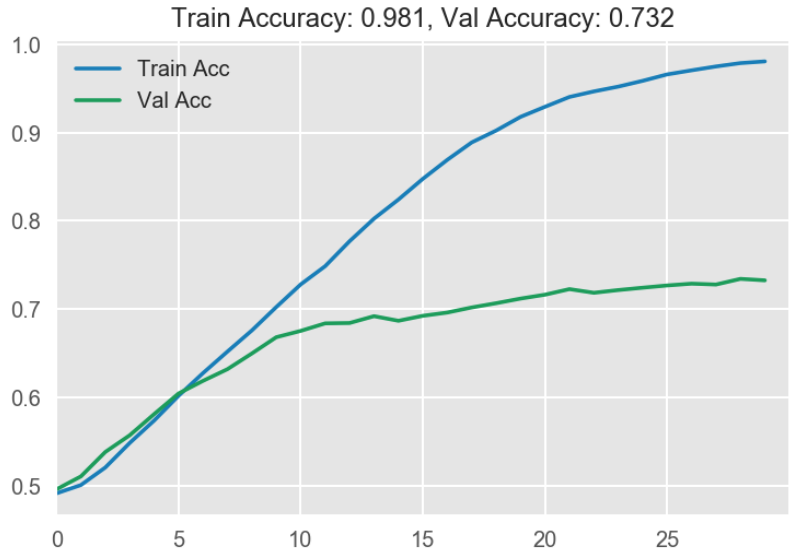
No Dropout

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

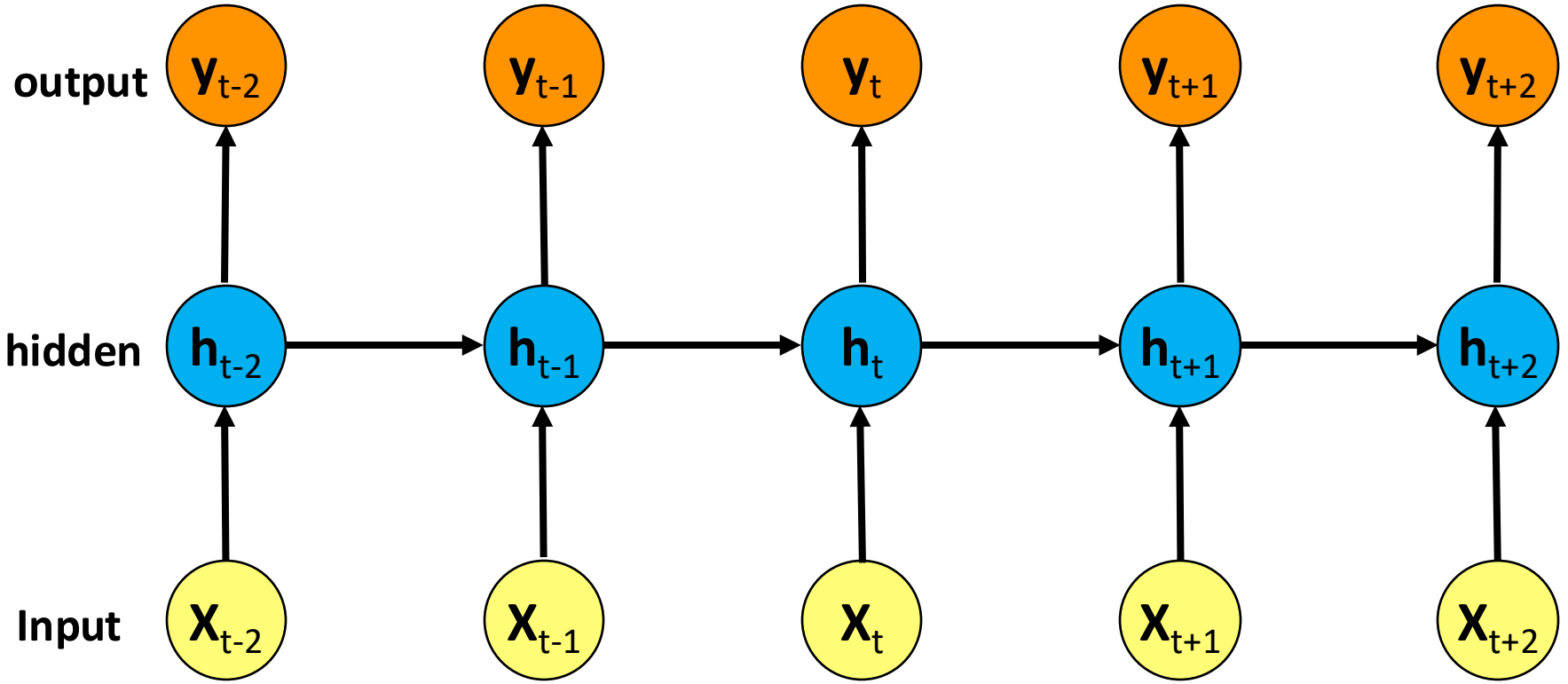




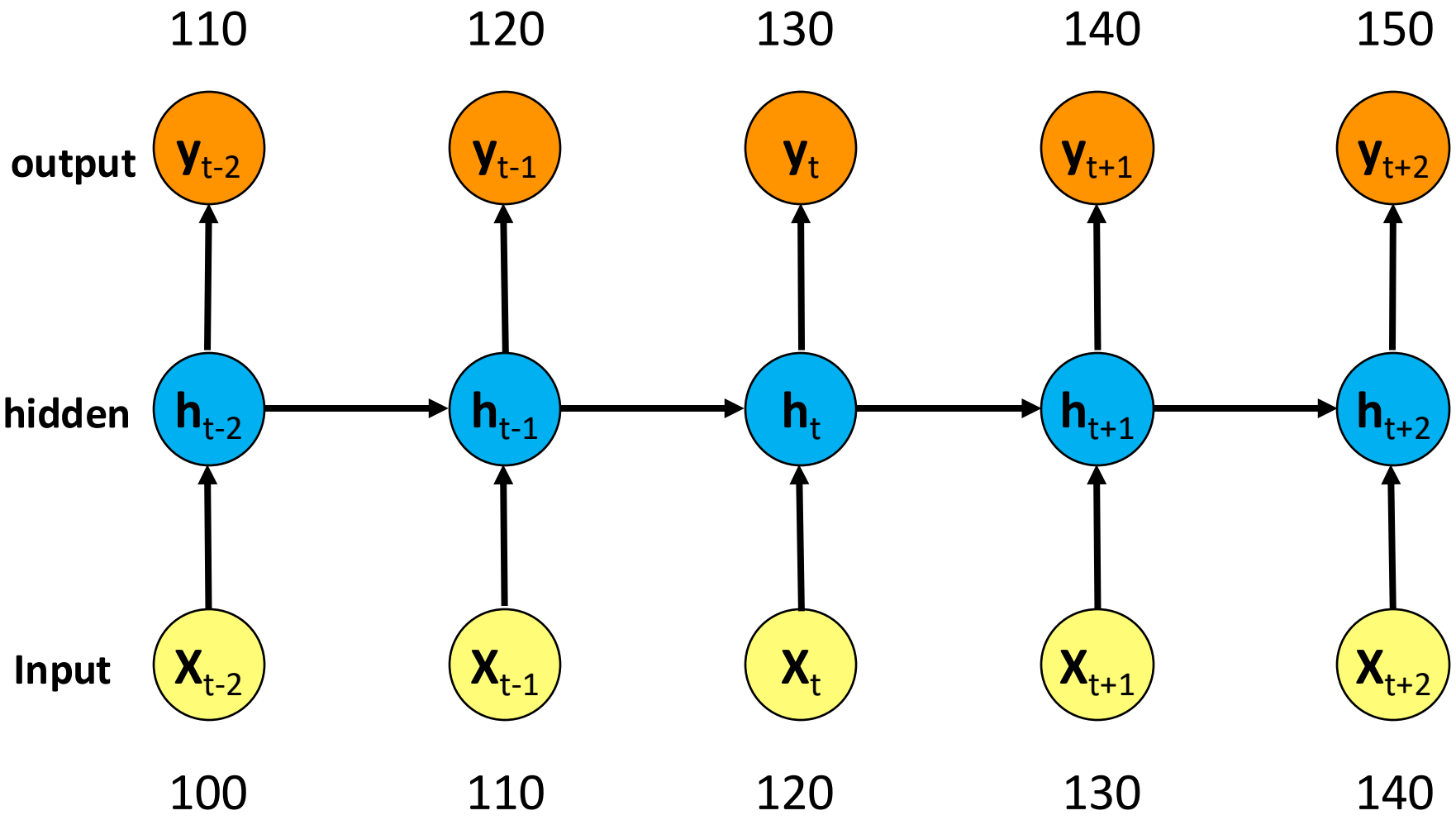
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Recurrent Neural Networks (RNN)

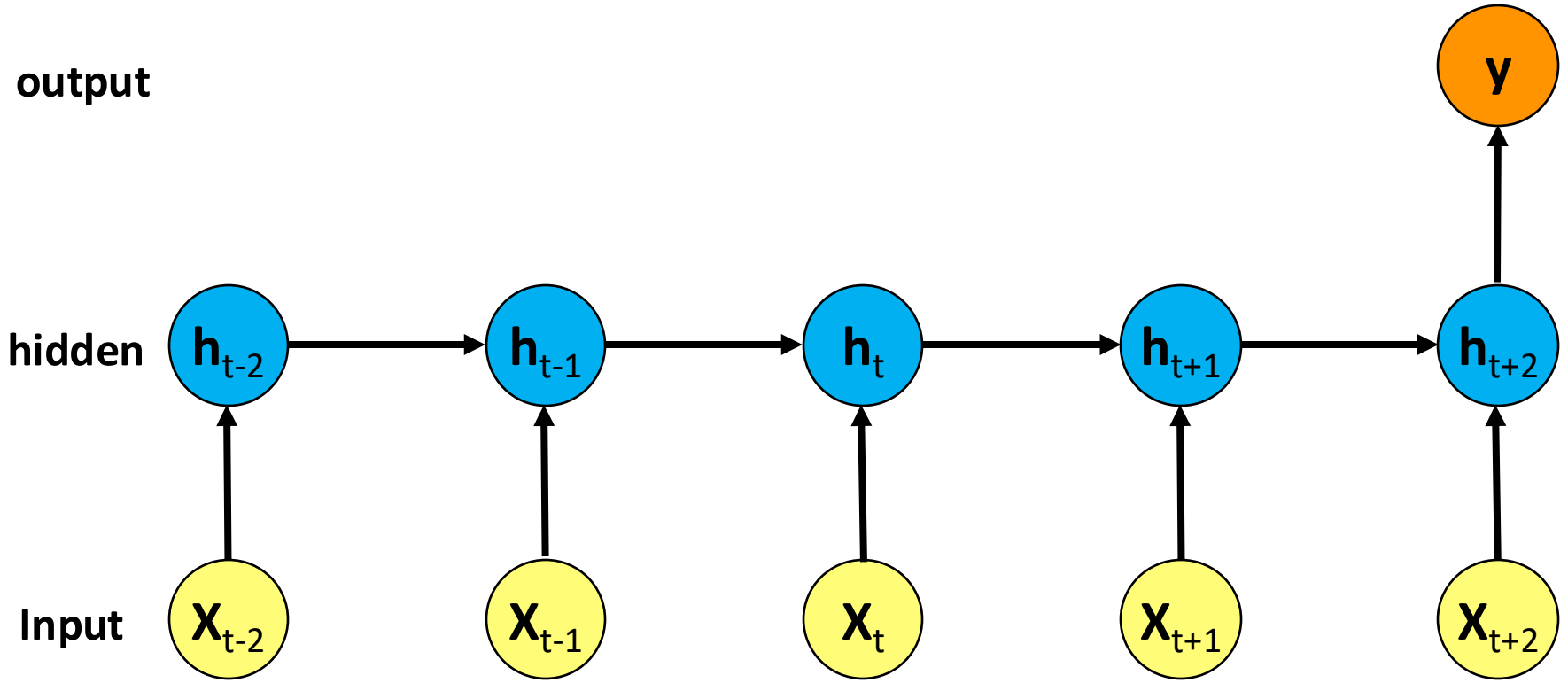
Recurrent Neural Networks (RNN)



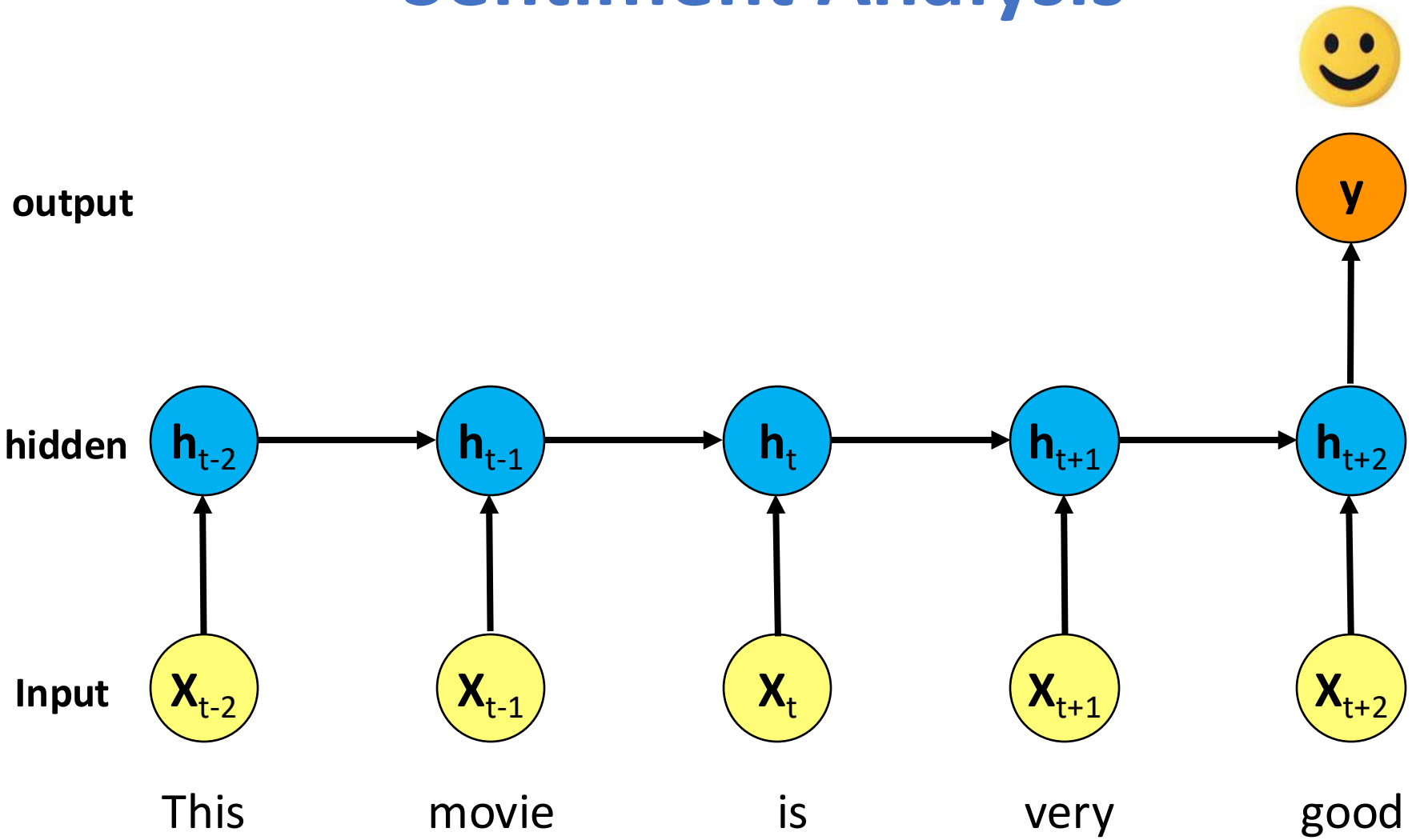
Recurrent Neural Networks (RNN)
Time Series Forecasting



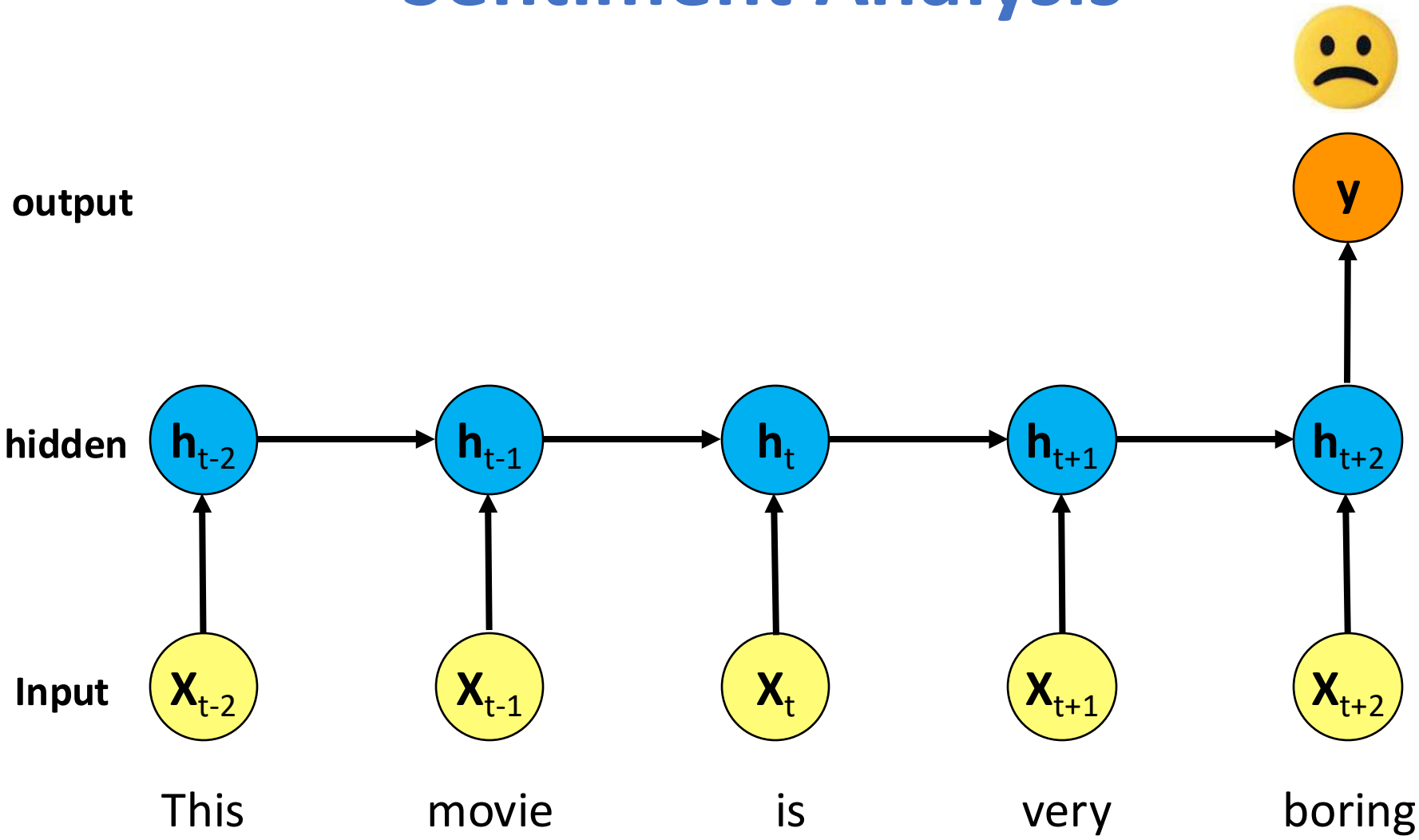
Recurrent Neural Networks (RNN)



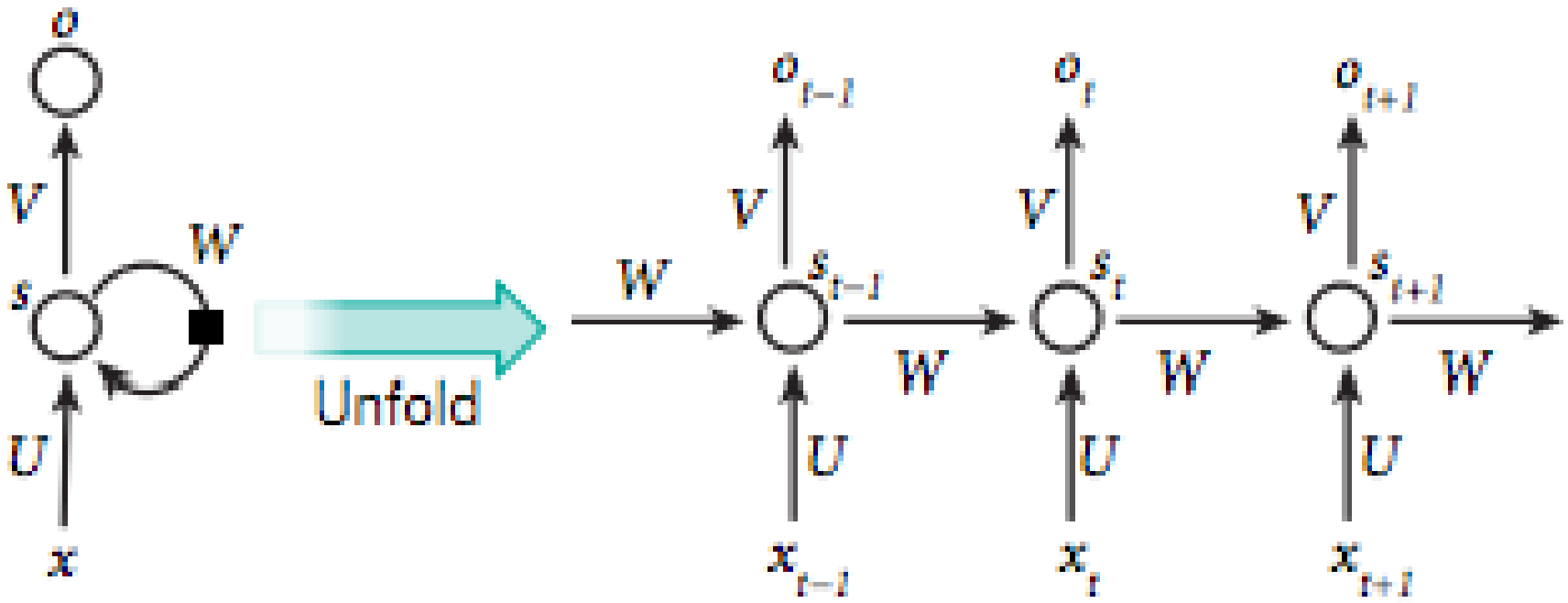
Recurrent Neural Networks (RNN)
Sentiment Analysis



Recurrent Neural Networks (RNN)
Sentiment Analysis

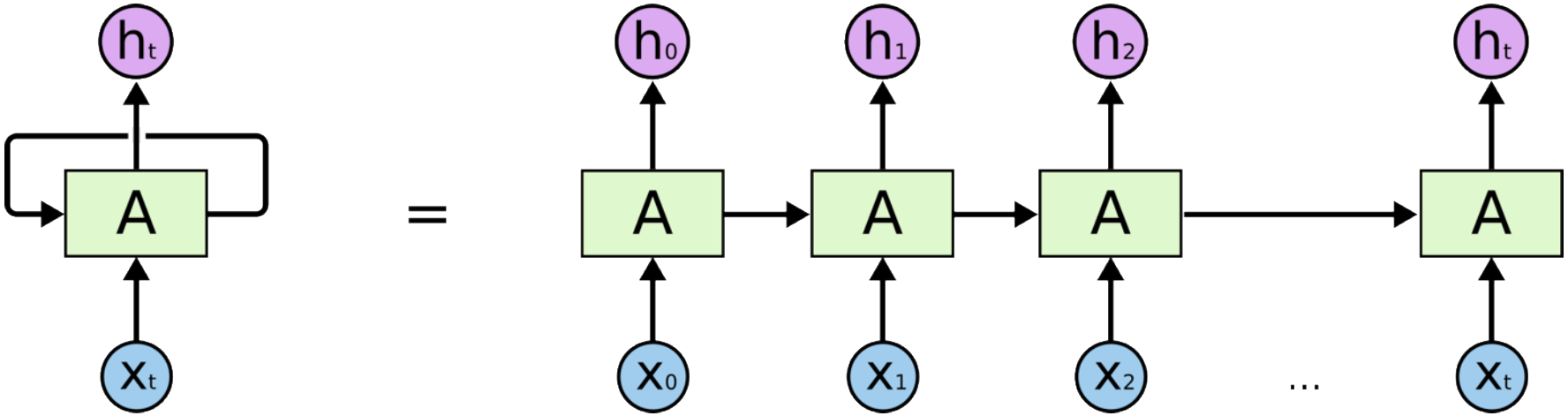


Recurrent Neural Network (RNN)



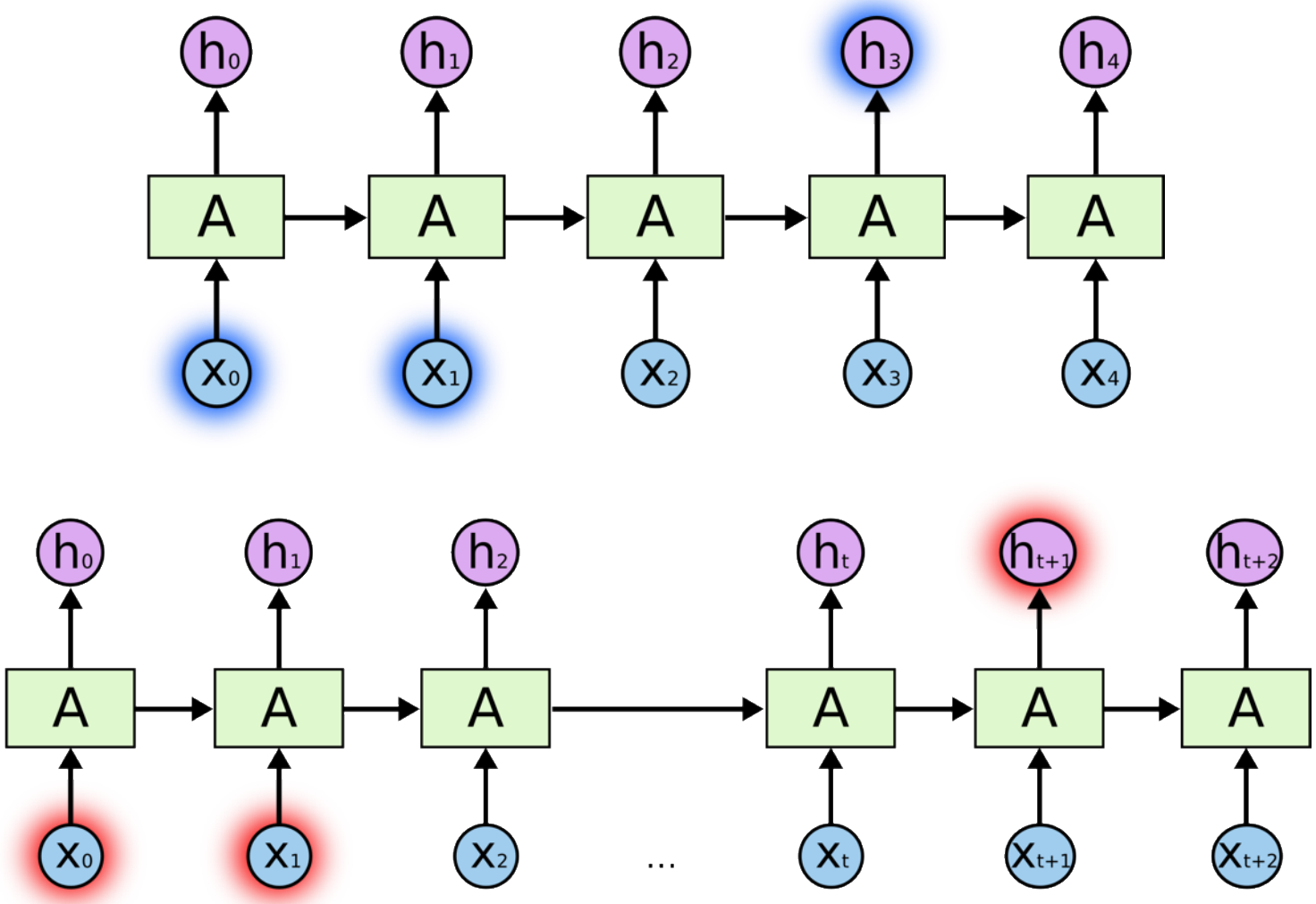
Source: LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521, no. 7553 (2015): 436-444.

RNN



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

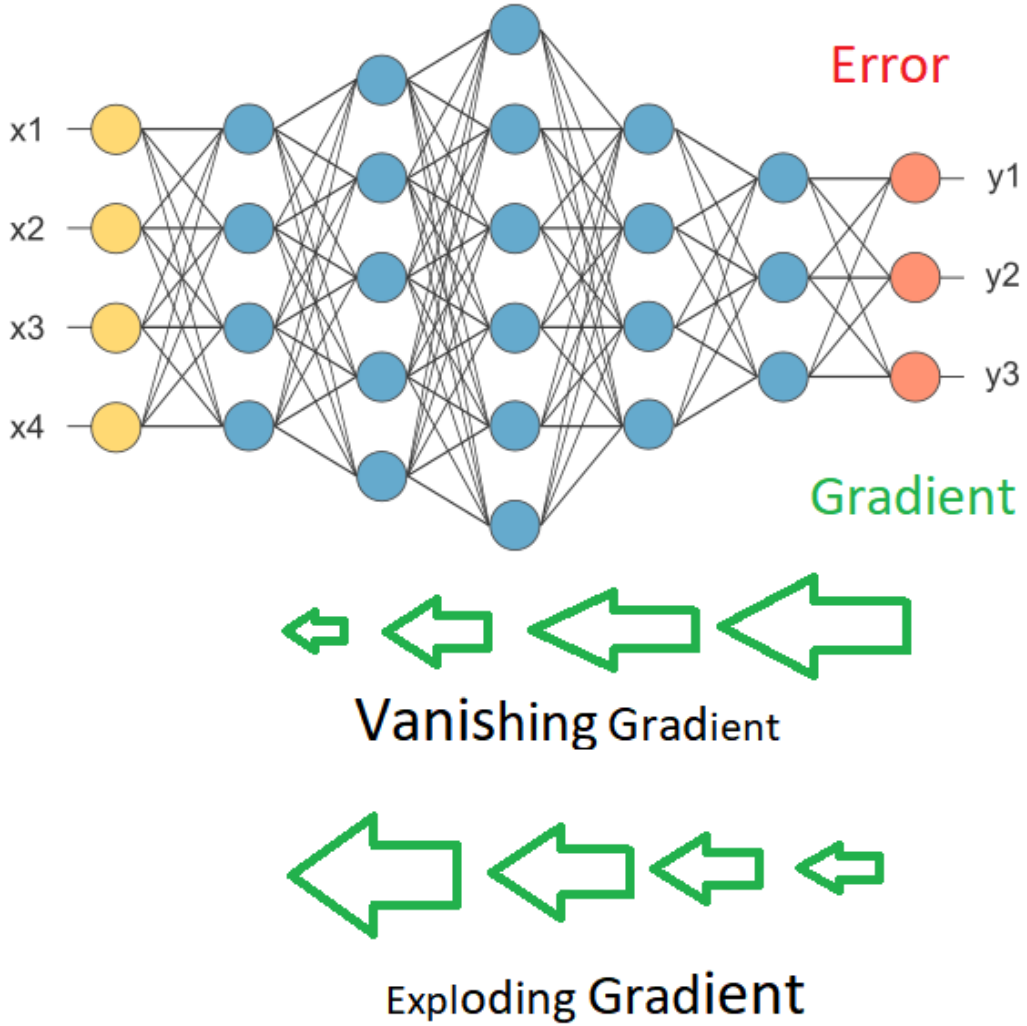
RNN long-term dependencies



I grew up in France… I speak fluent French.

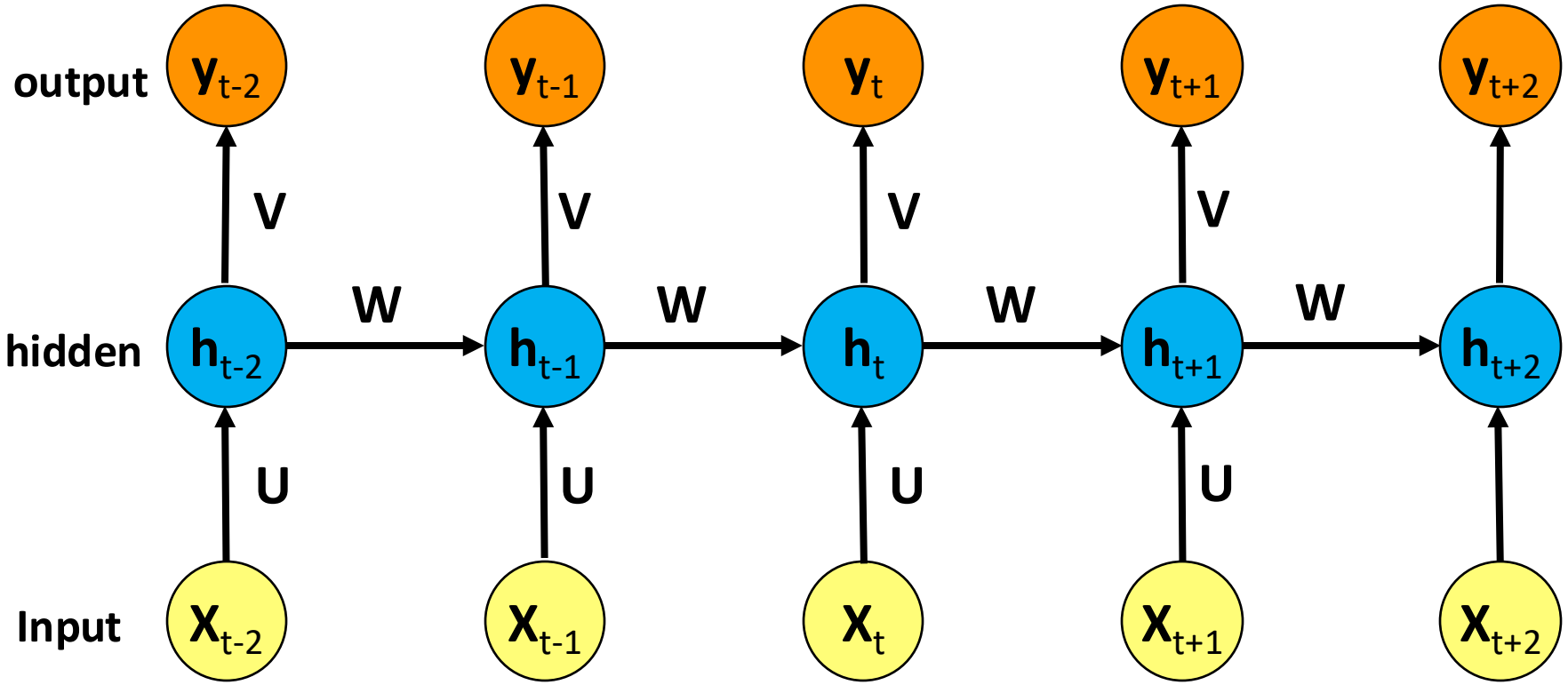
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Vanishing Gradient
Exploding Gradient

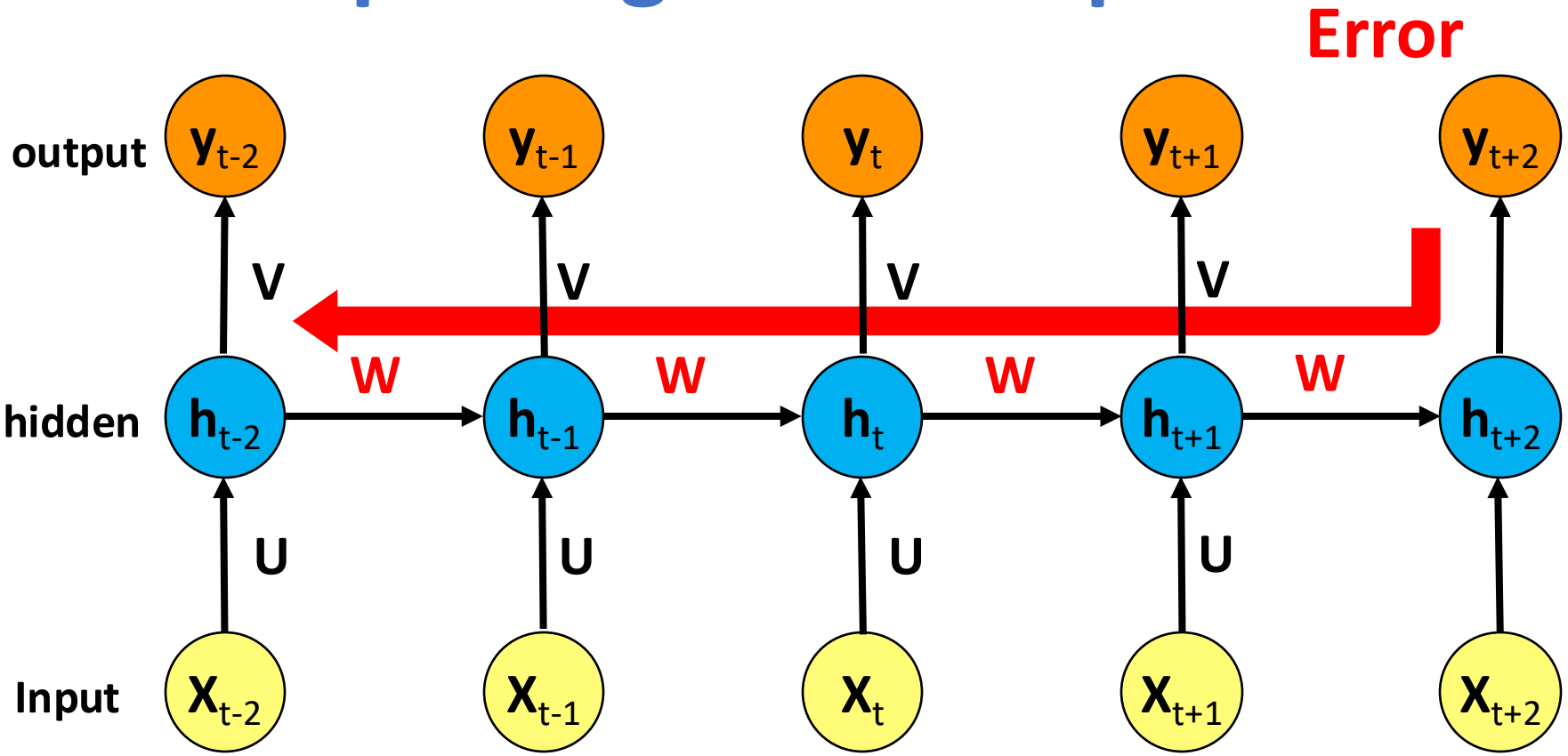


Source: https://medium.com/deep-math-machine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235

Recurrent Neural Networks (RNN)



RNN
Vanishing Gradient problem
Exploding Gradient problem

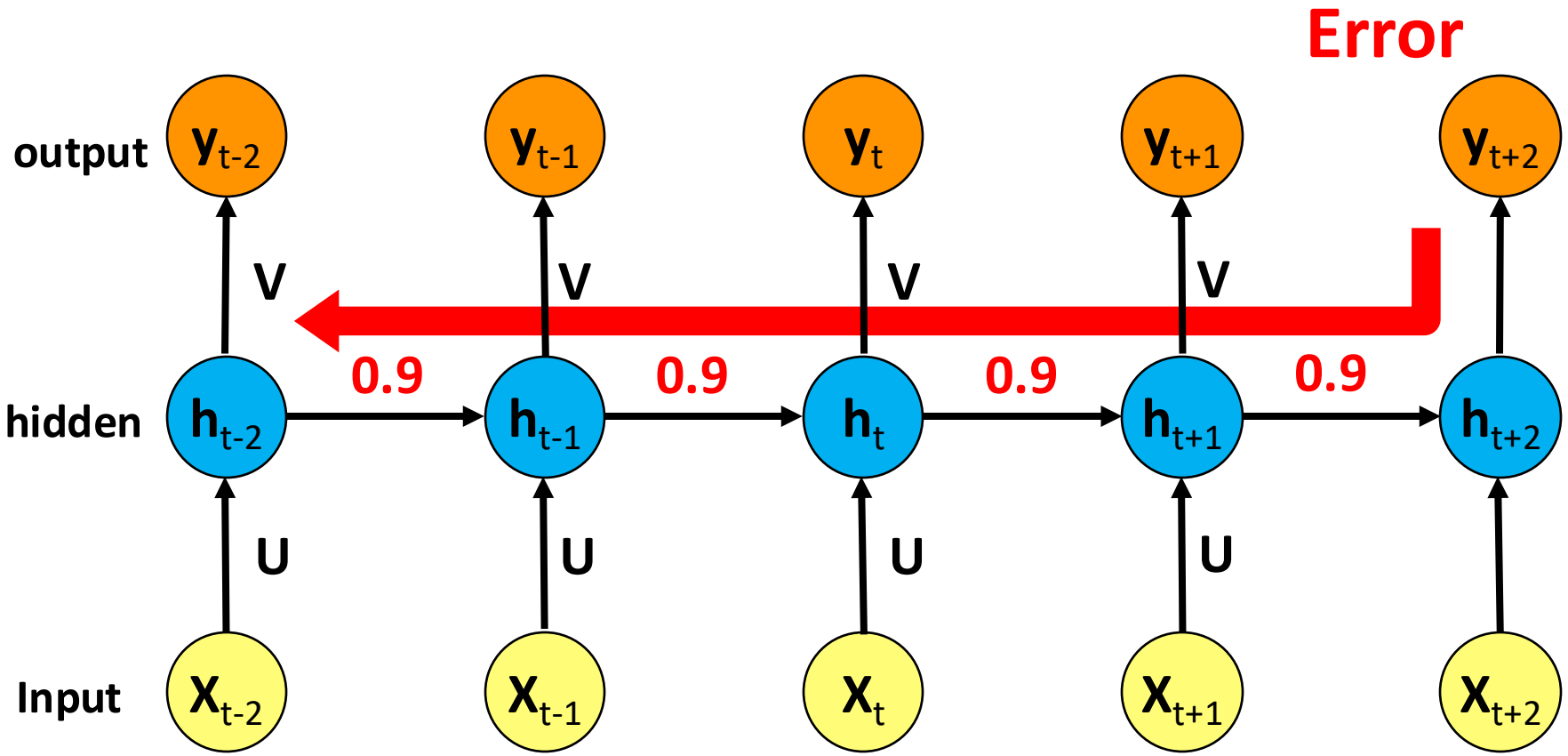


if |W| < 1 (Vanishing)

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

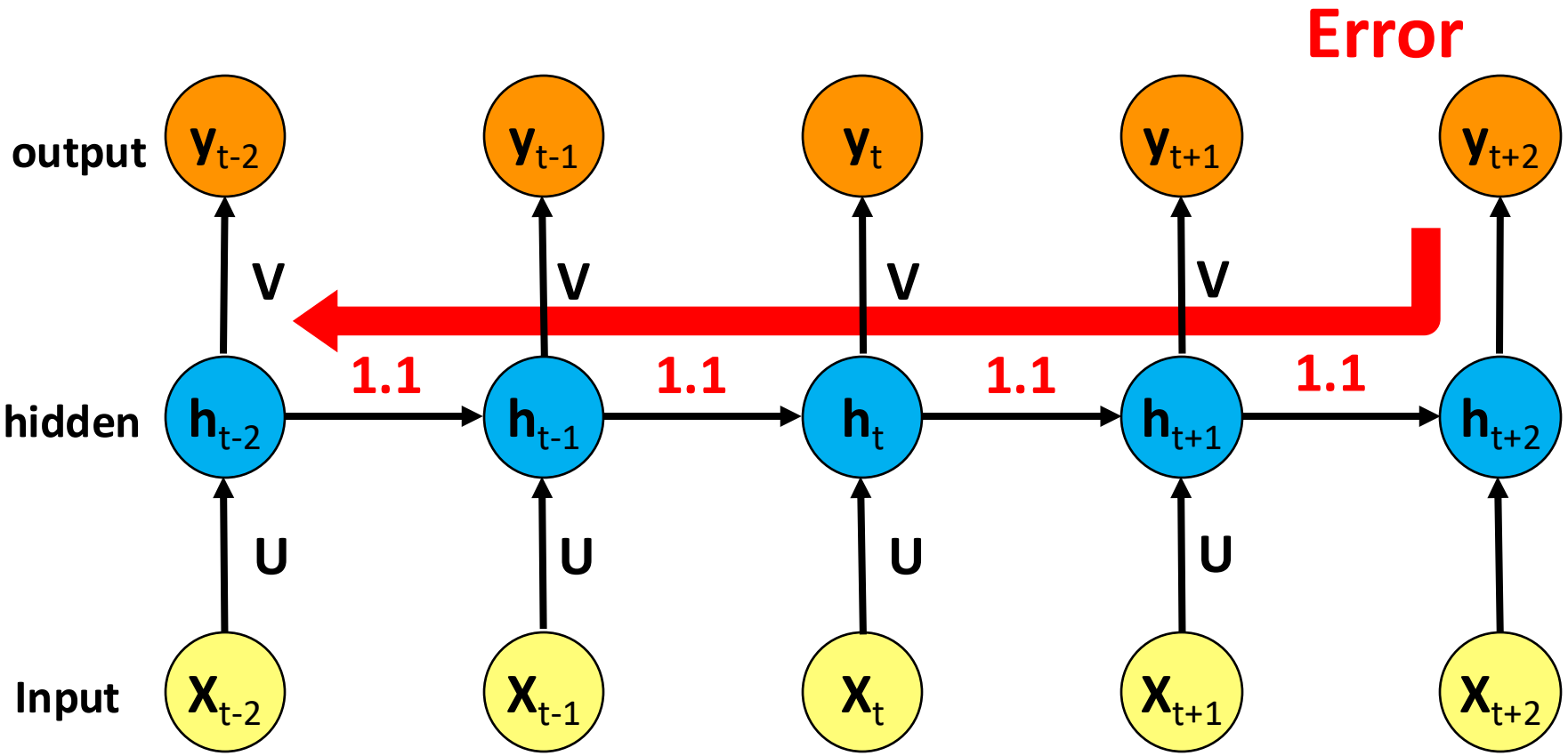
RNN
Vanishing Gradient problem



W = 0.9 < 1 (Vanishing)

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

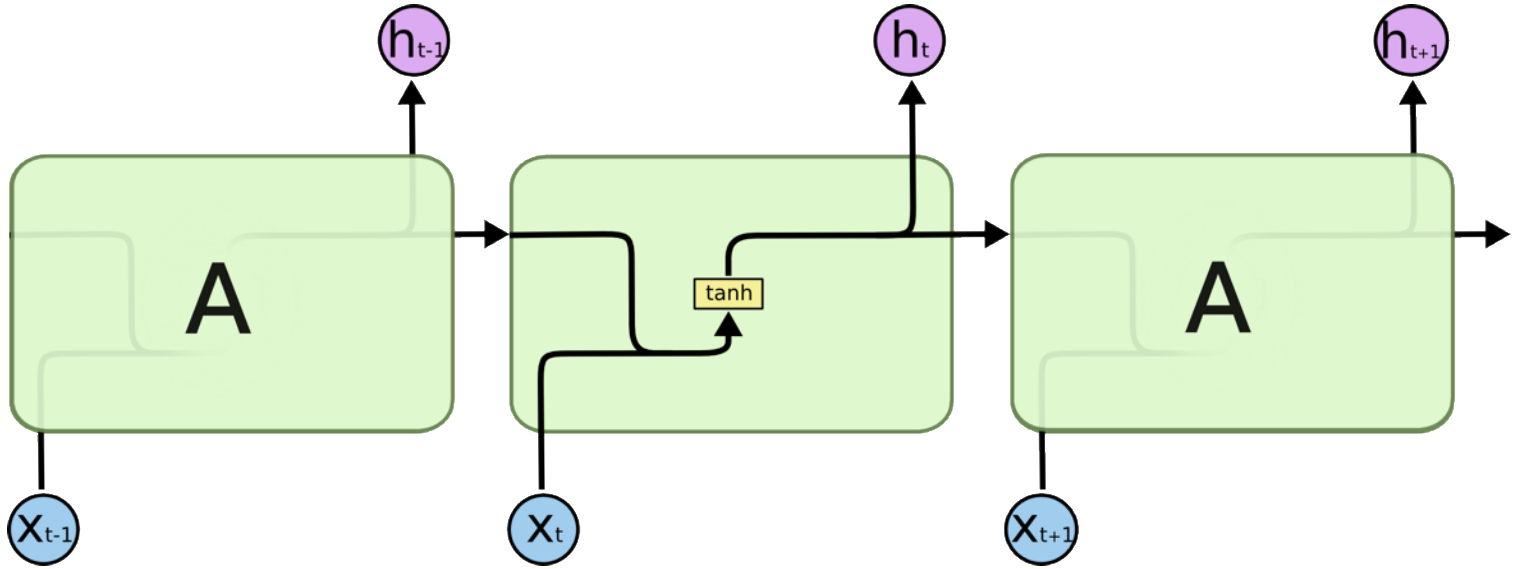


W = 1.1 > 1 (Exploding)

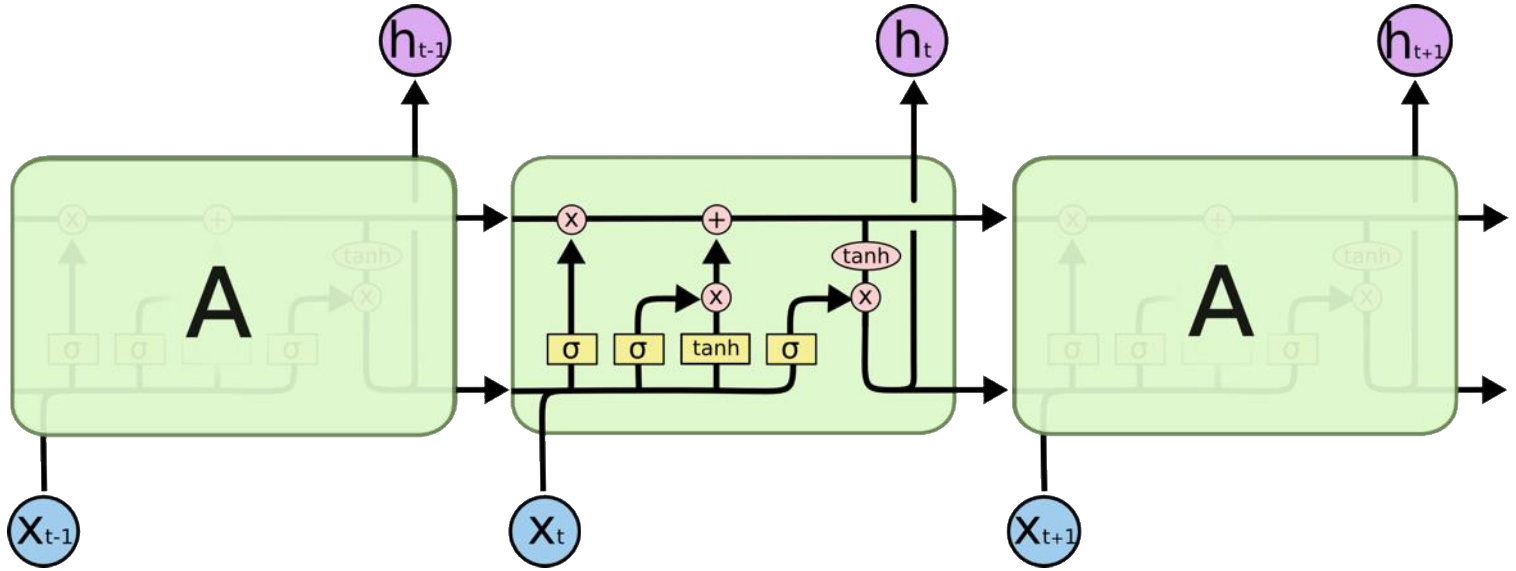
Source: https://medium.com/deep-math-machine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235

RNN LSTM

RNN

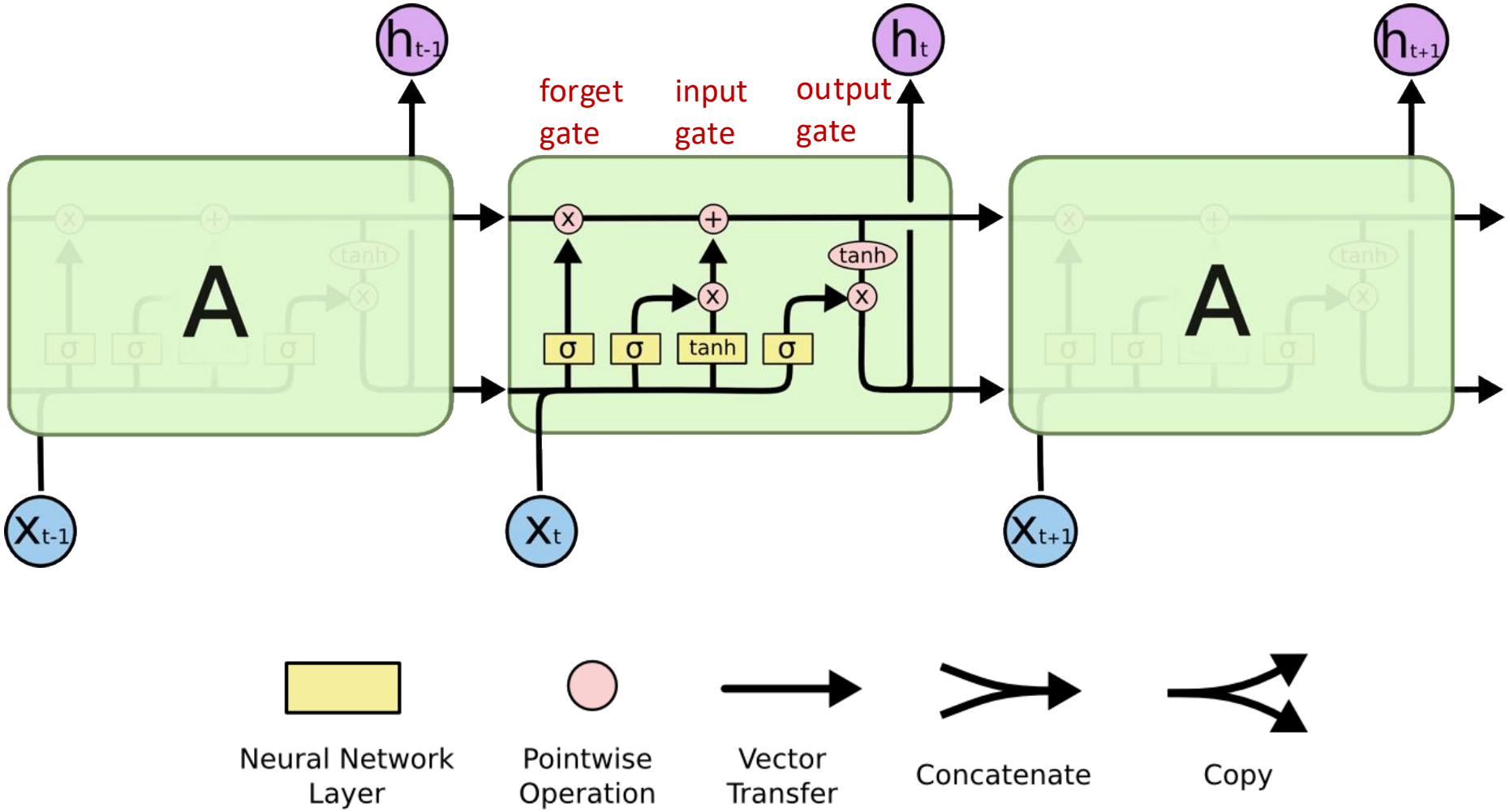


LSTM



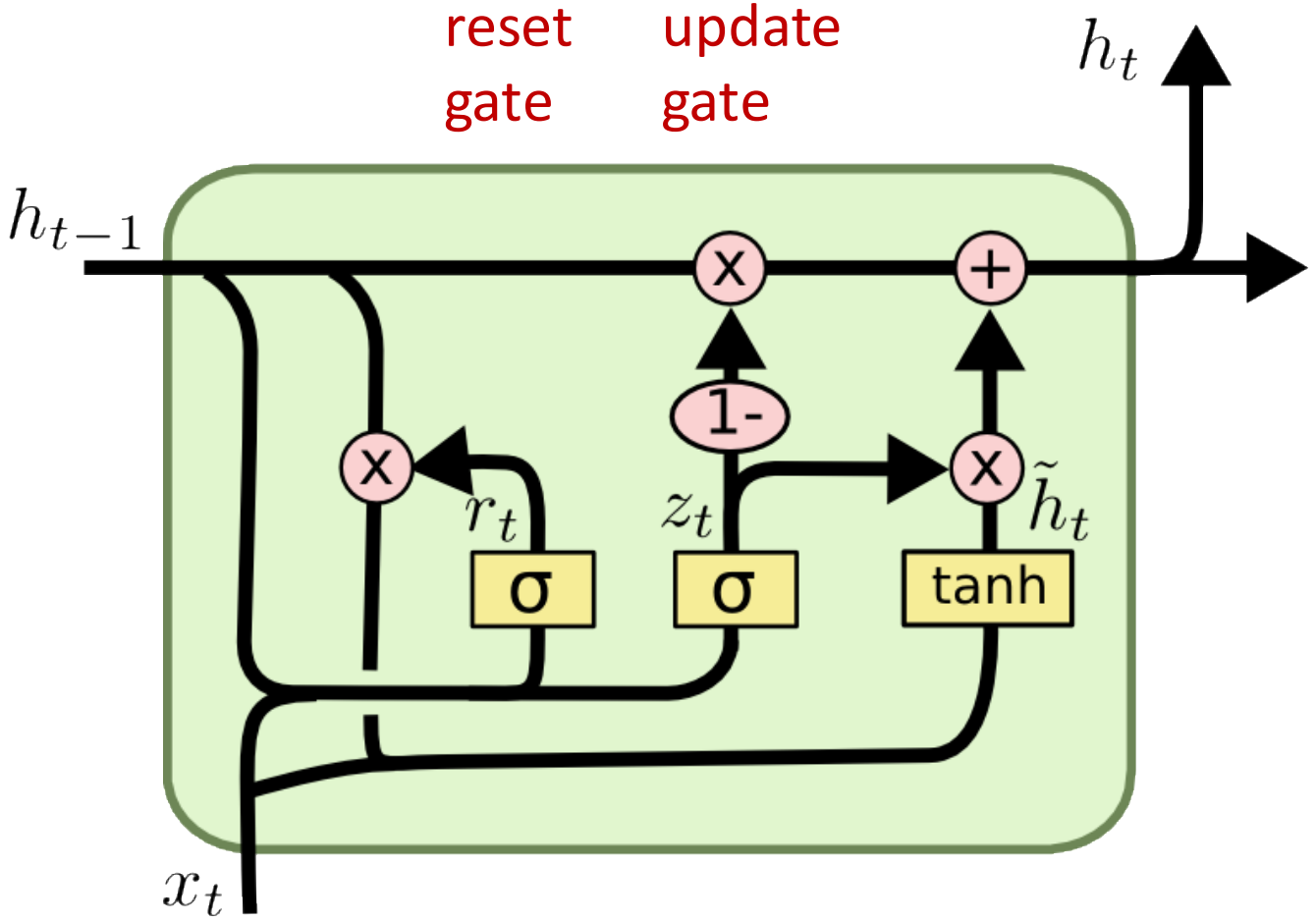
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory (LSTM)



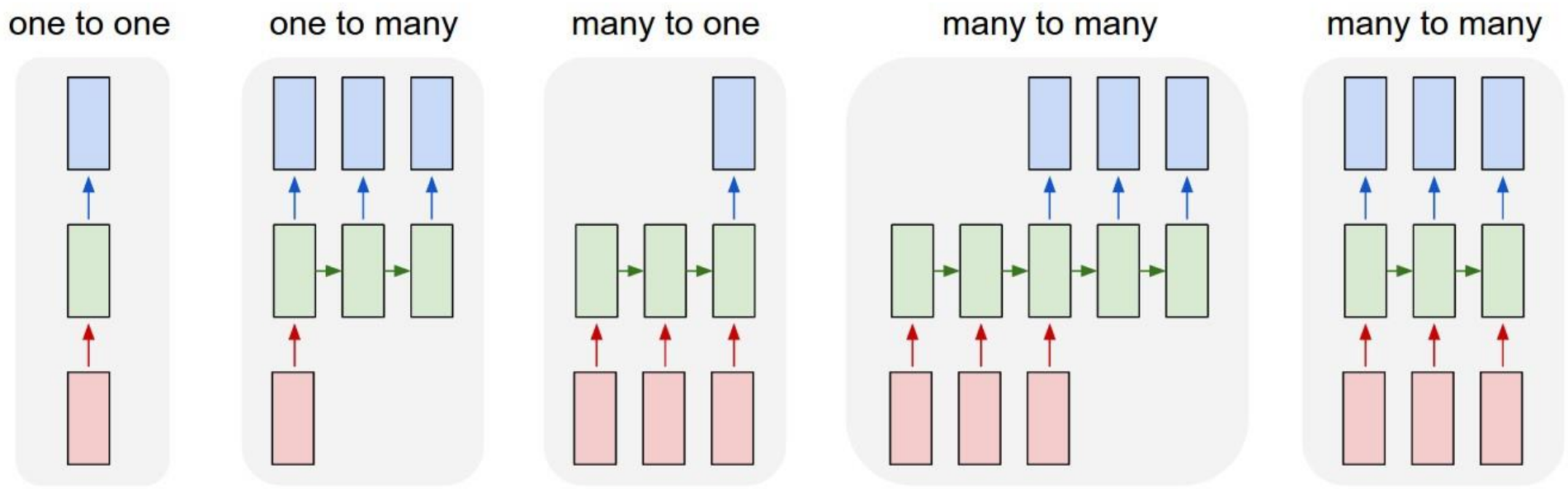
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

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

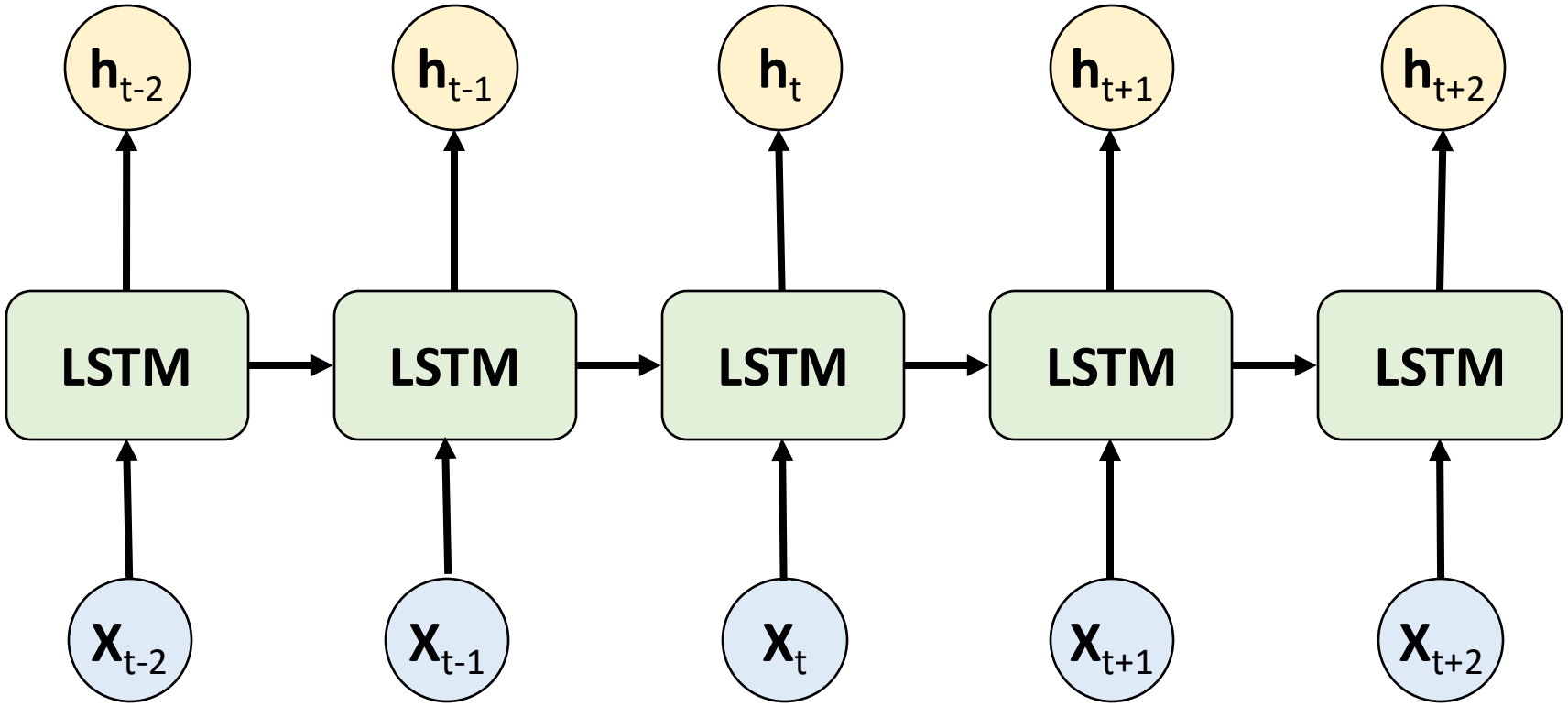
Sentiment Classification

Name Entity Recognition

Machine Translation

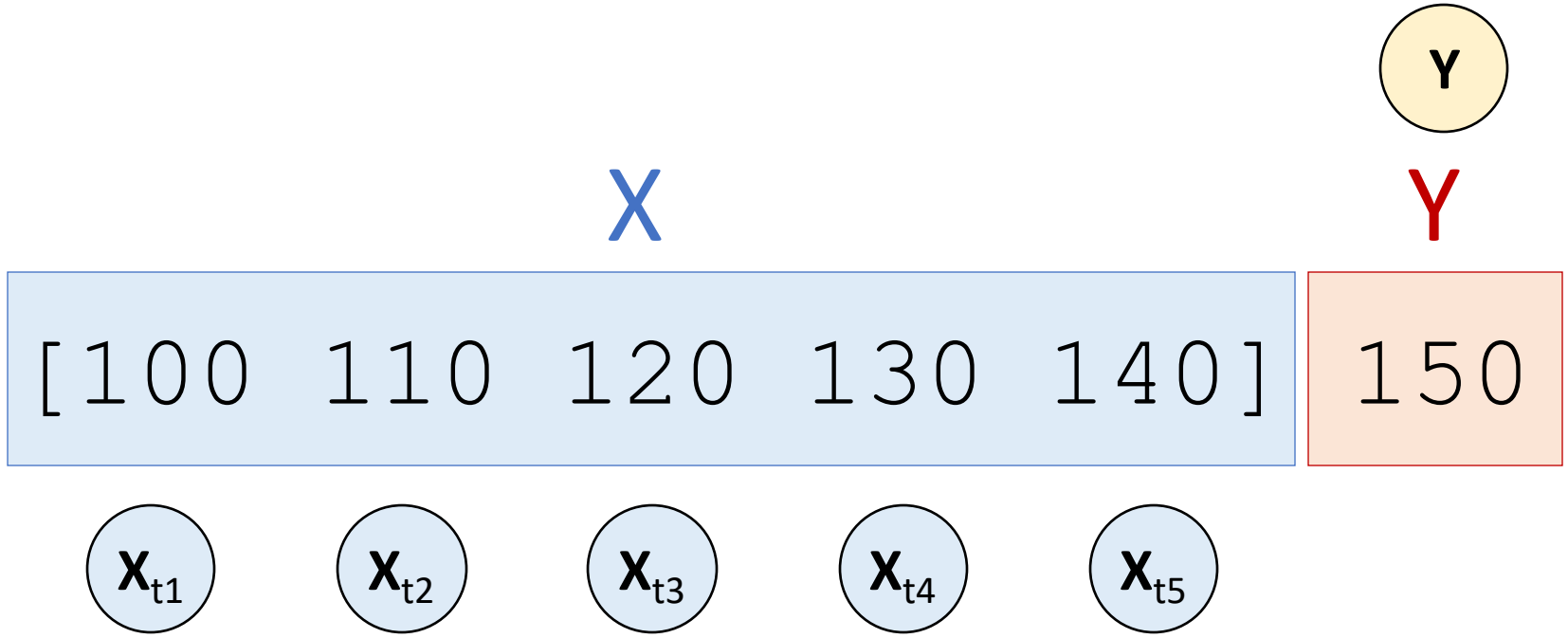
Source: https://github.com/Vict0rSch/deep_learning/tree/master/keras/recurrent

Long Short Term Memory (LSTM) for Time Series Forecasting



Time Series Data

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

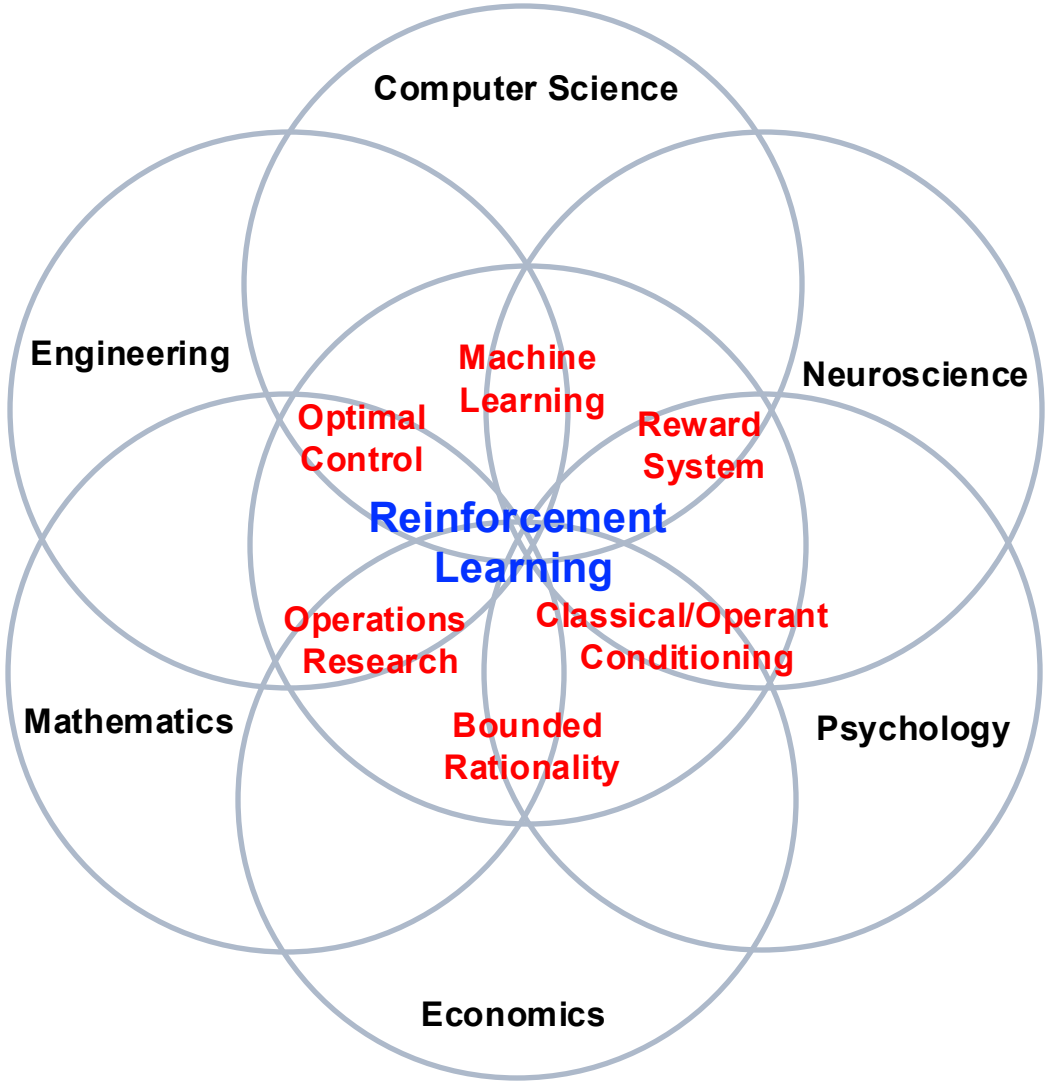


Time Series Data

[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

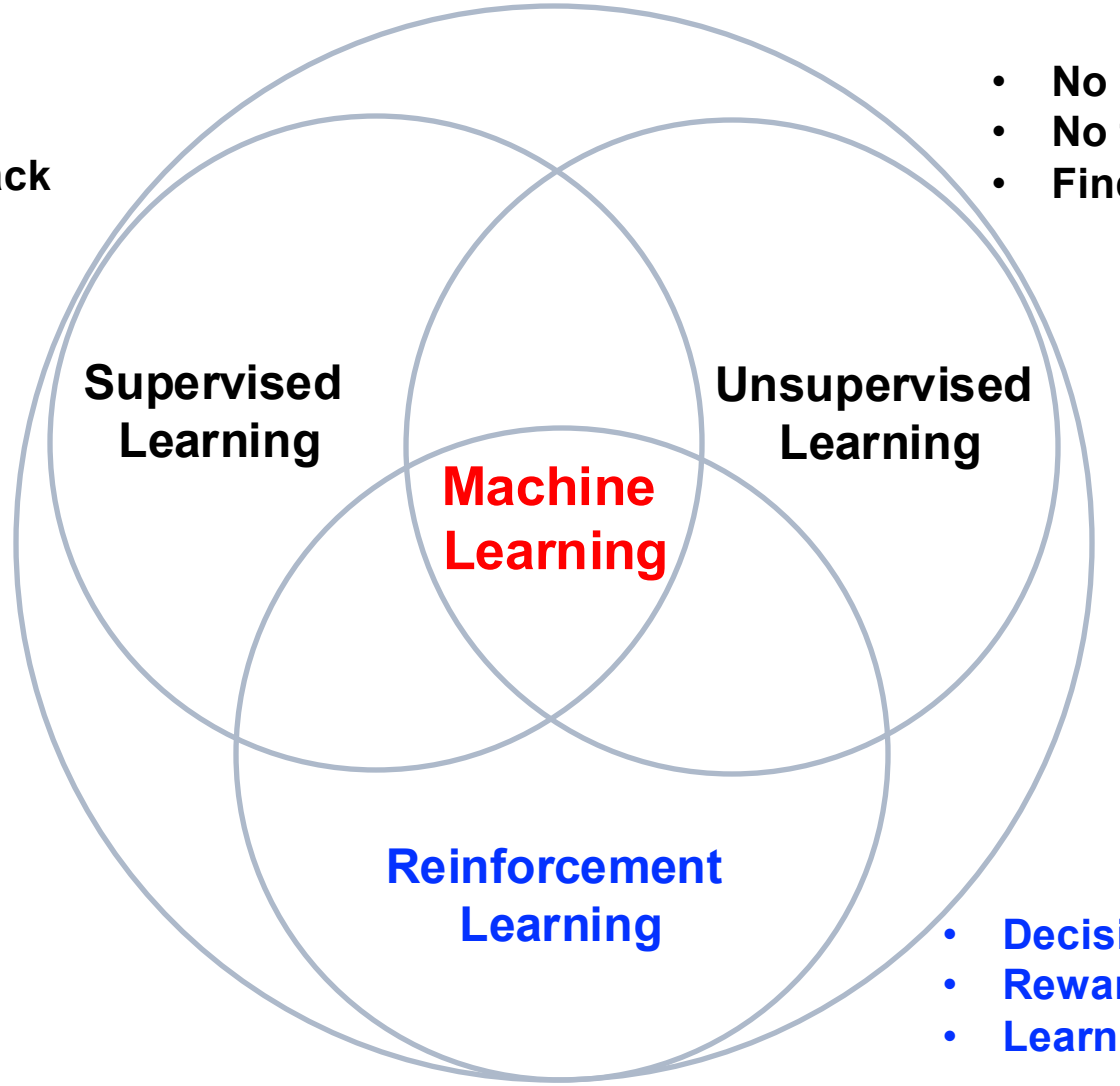
Reinforcement Learning (RL)



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

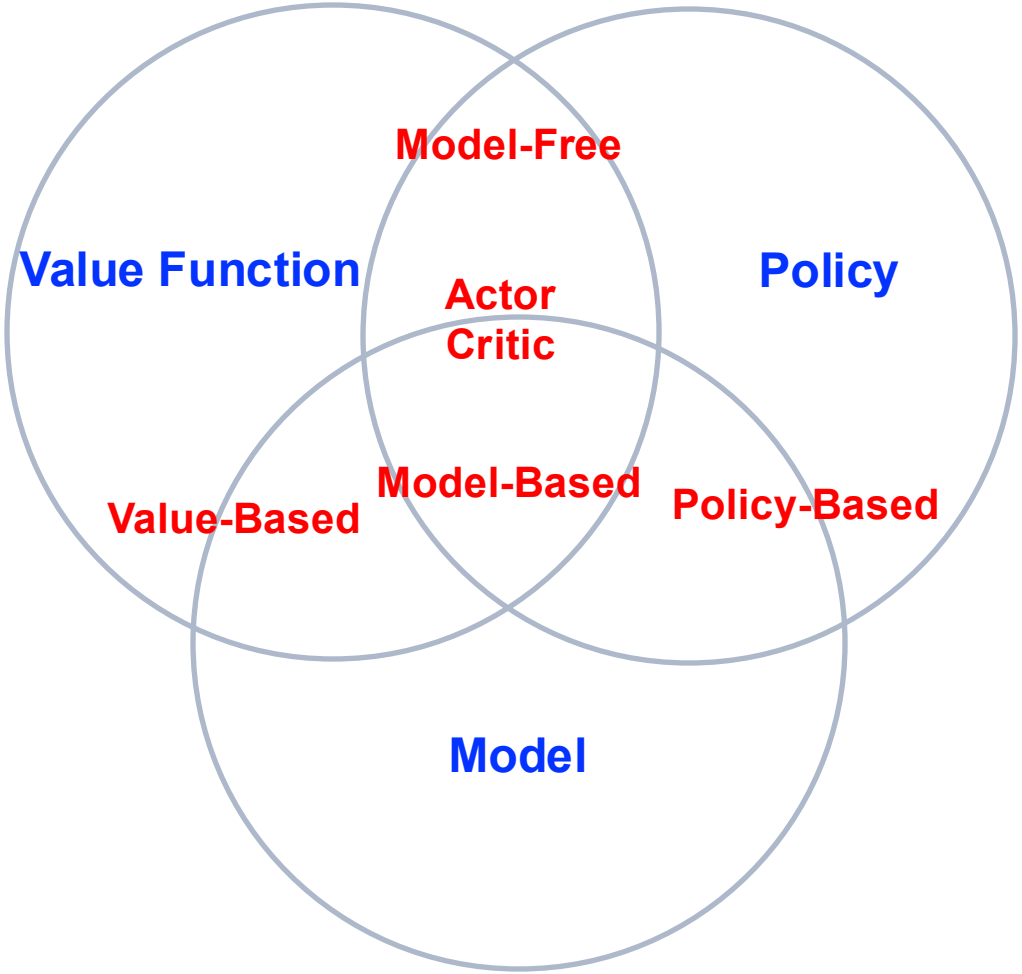


- No Labels
- No feedback
- Find hidden structure

- Decision process
- Reward system
- Learn series of actions

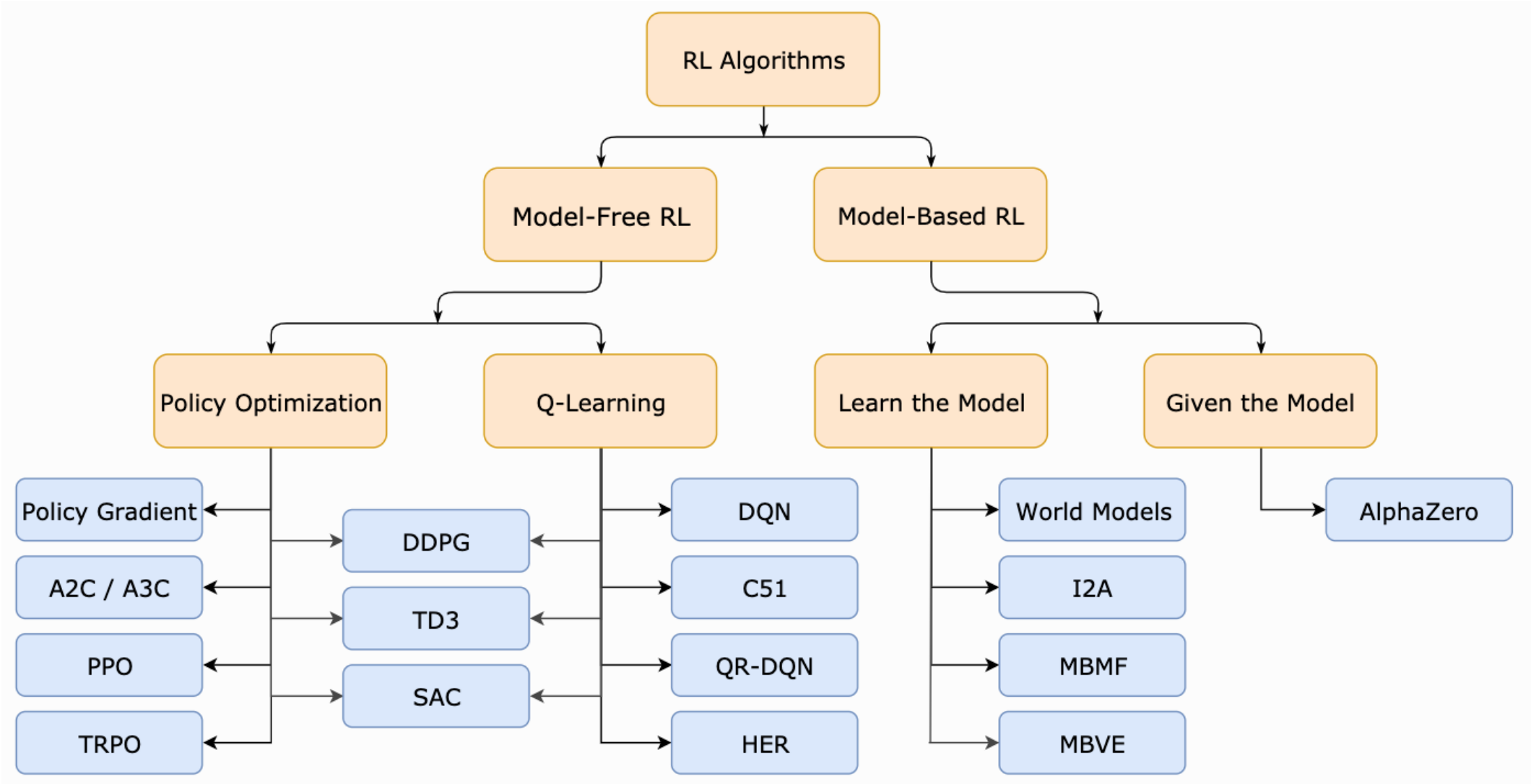
Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning (RL) Taxonomy



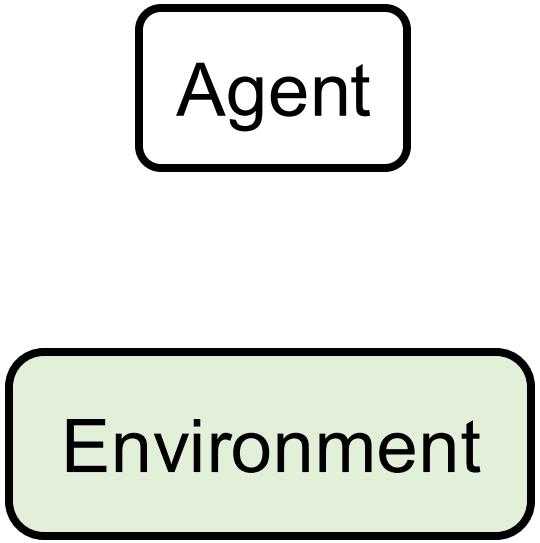
Source: David Silver (2015), Introduction to reinforcement learning, https://www.youtube.com/playlist?list=PLqYmG7hTraZDM-OYHWgPebj2MfCFzFObQ

Reinforcement Learning (RL)
A Taxonomy of RL Algorithms



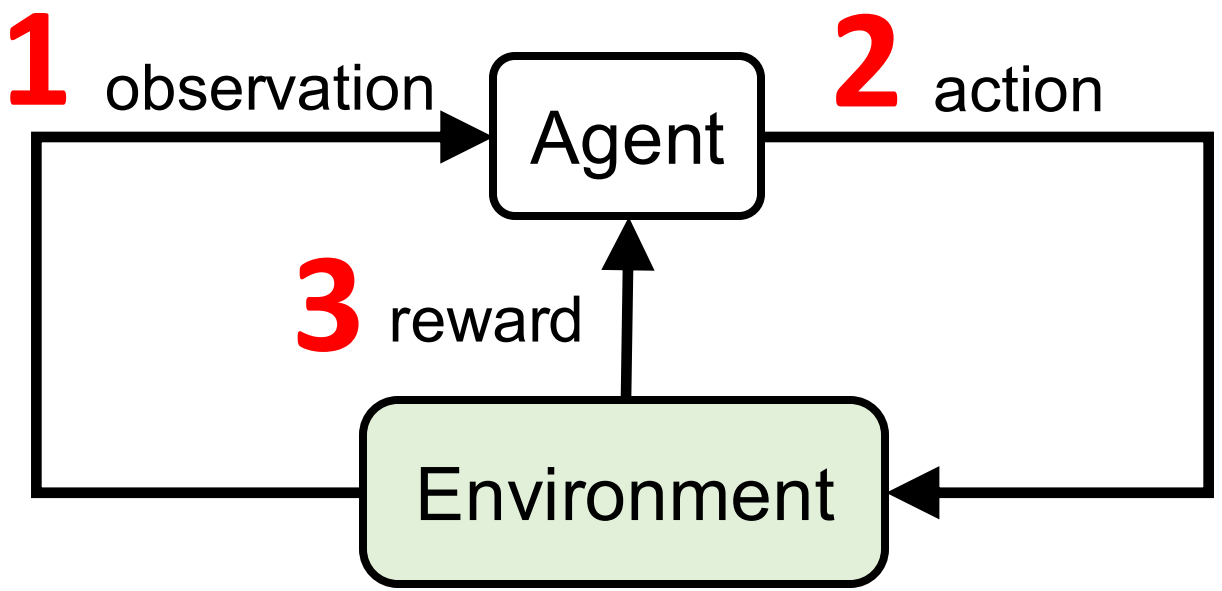
Source: https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html

Reinforcement Learning (DL)



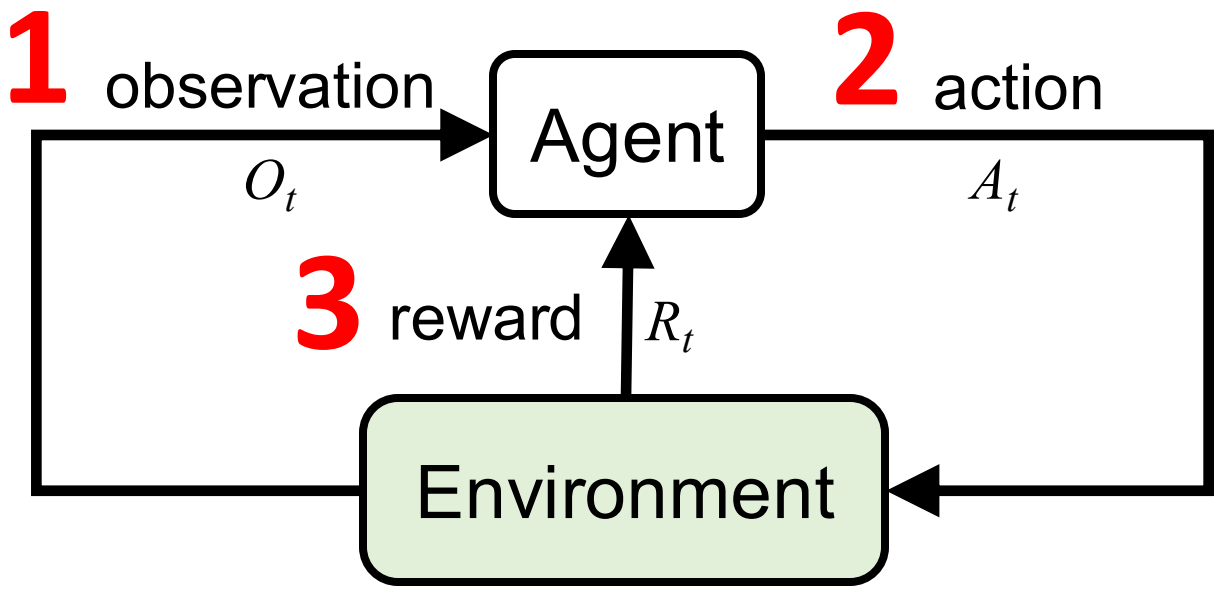
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning (DL)



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

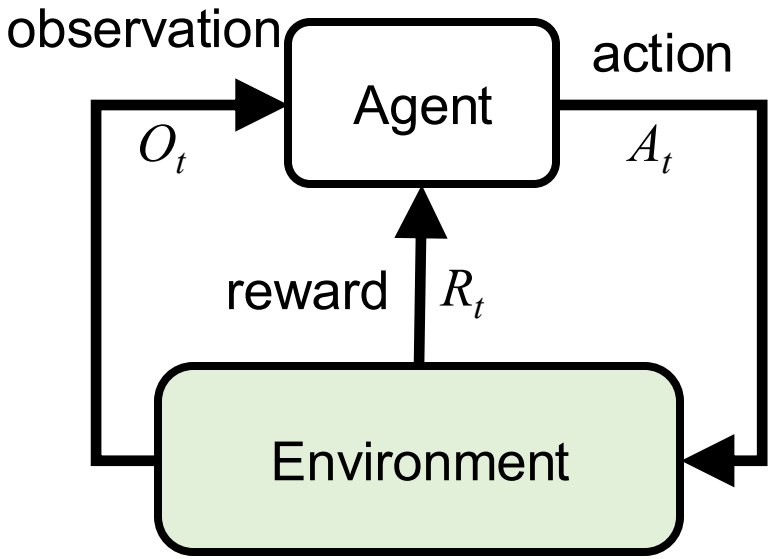
Reinforcement Learning (DL)



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

Agent and Environment

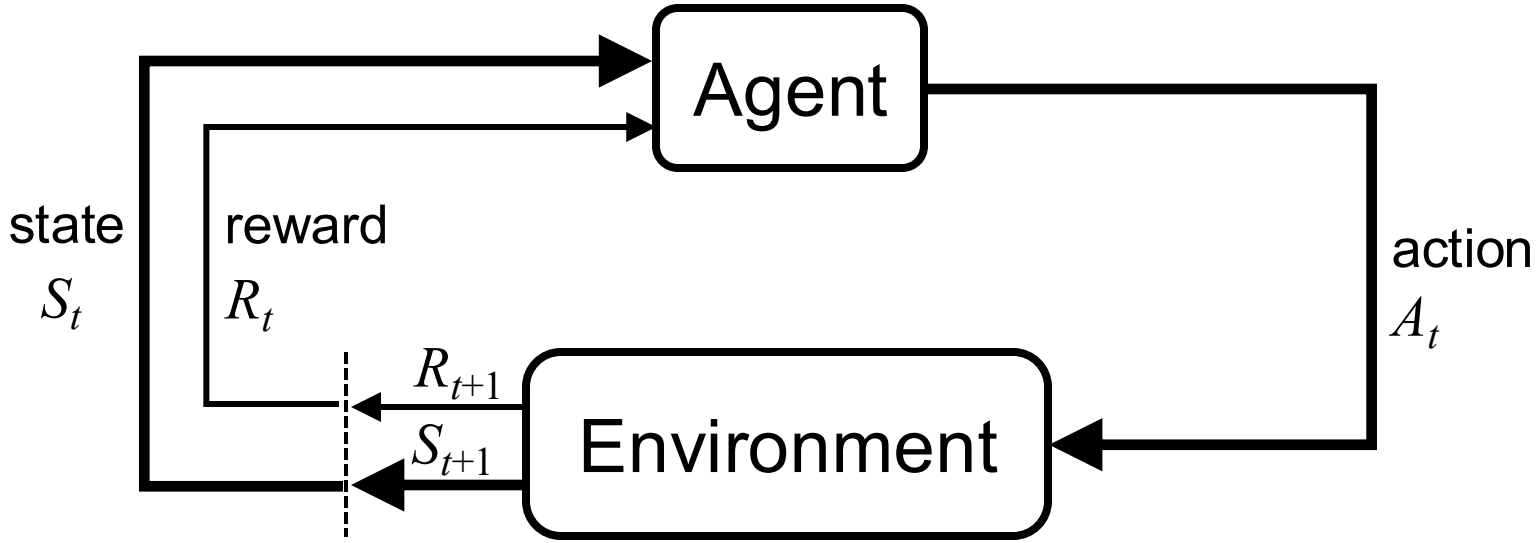
- At each step *t* the agent:
  - Executes action A_t
  - Receives observation O_t
  - Receives scalar reward R_t
- The environment:
  - Receives action A_t
  - Emits observation O_{t+1}
  - Emits scalar reward R_{t+1}
- t increments at env. step



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

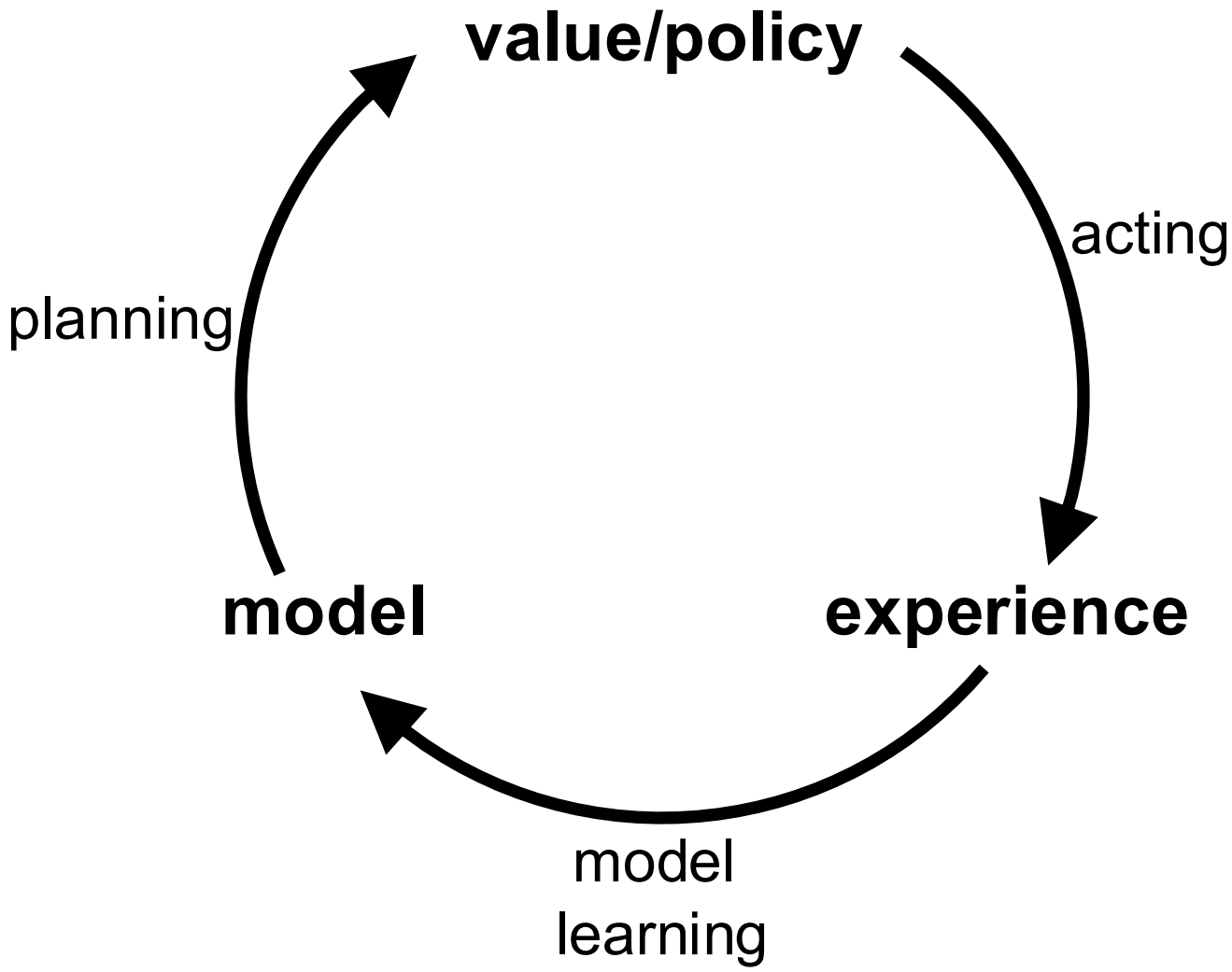
Reinforcement Learning (DL)

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



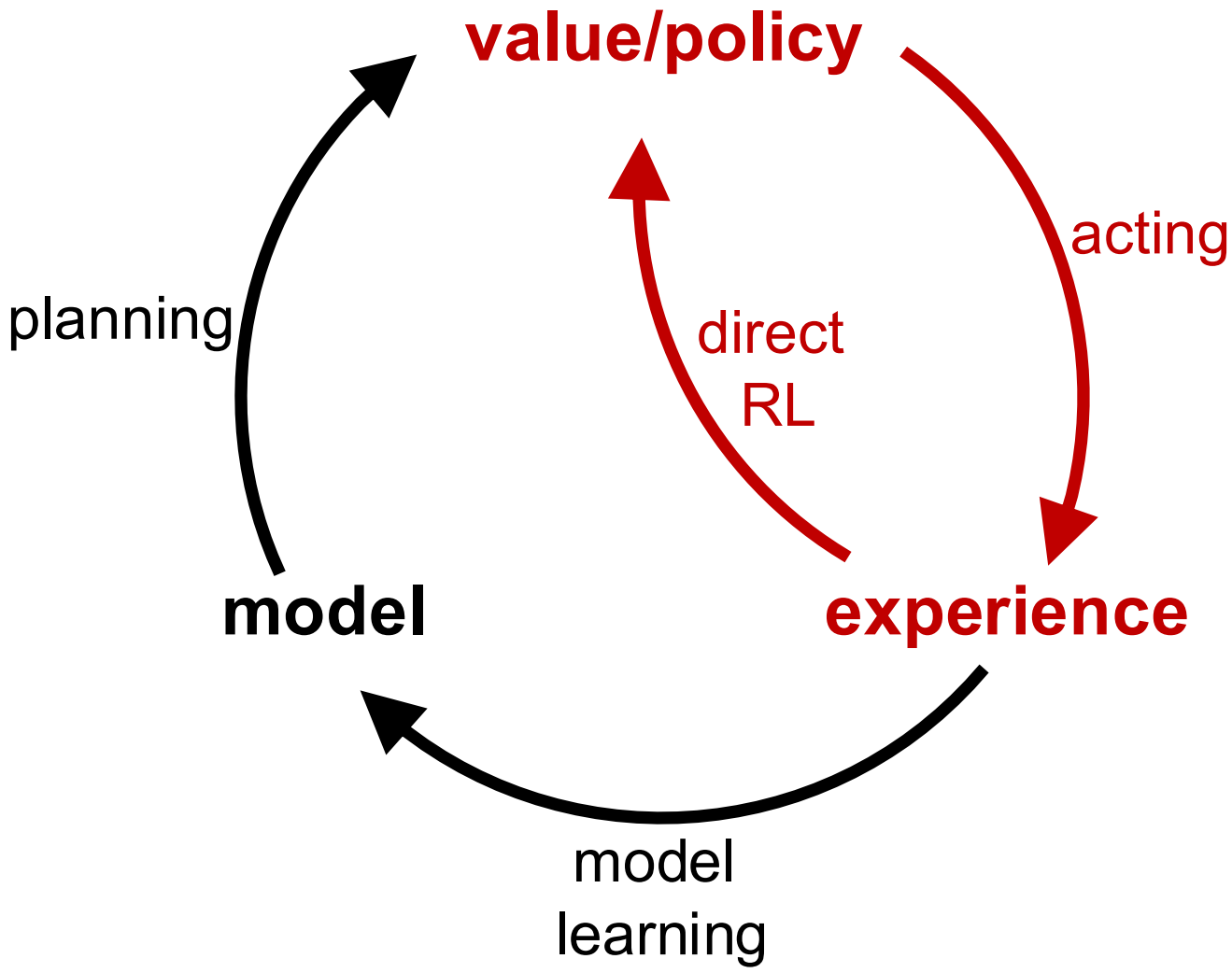
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Model-Based RL



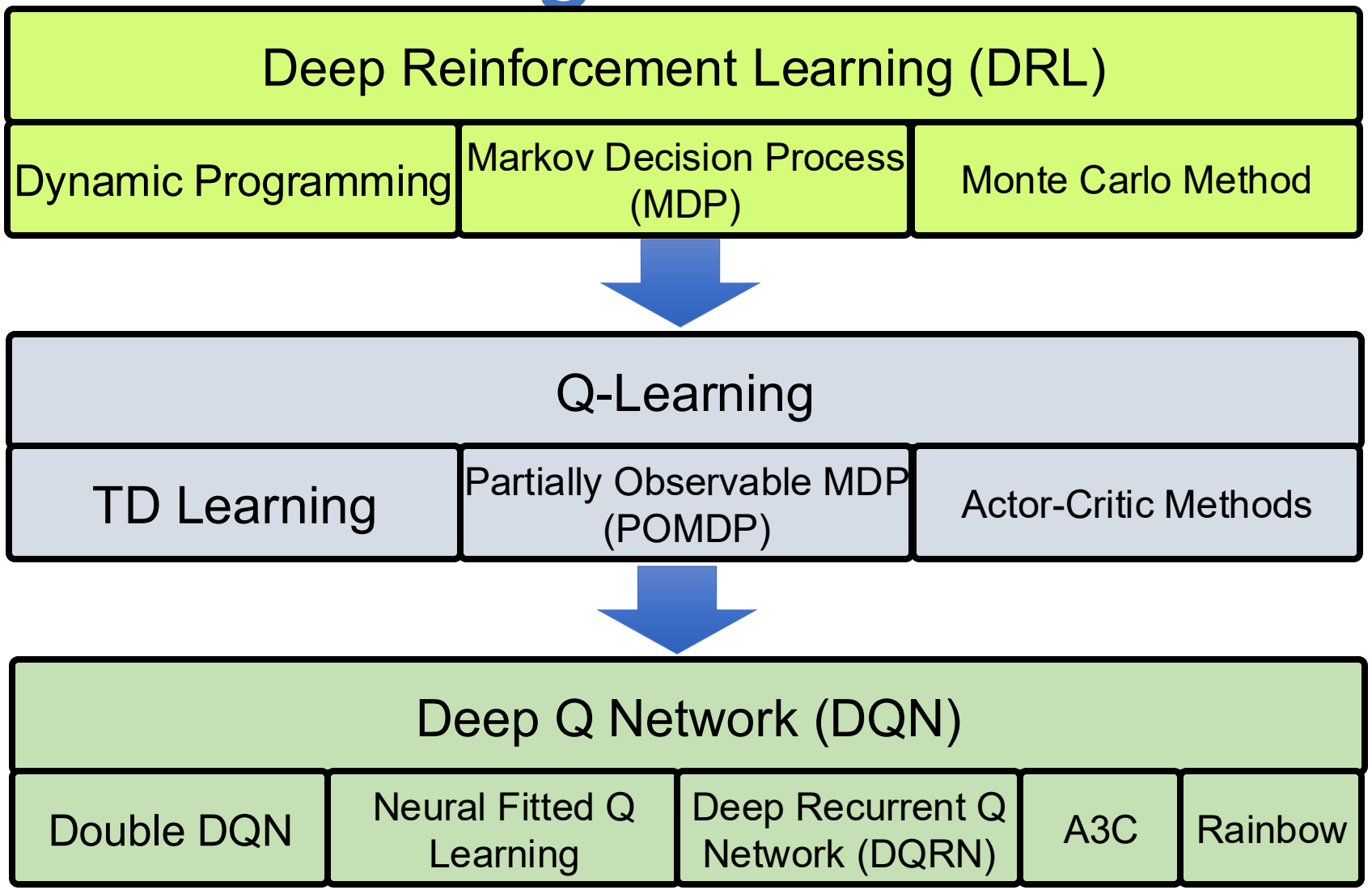
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Model-Free RL (DQN, A3C)



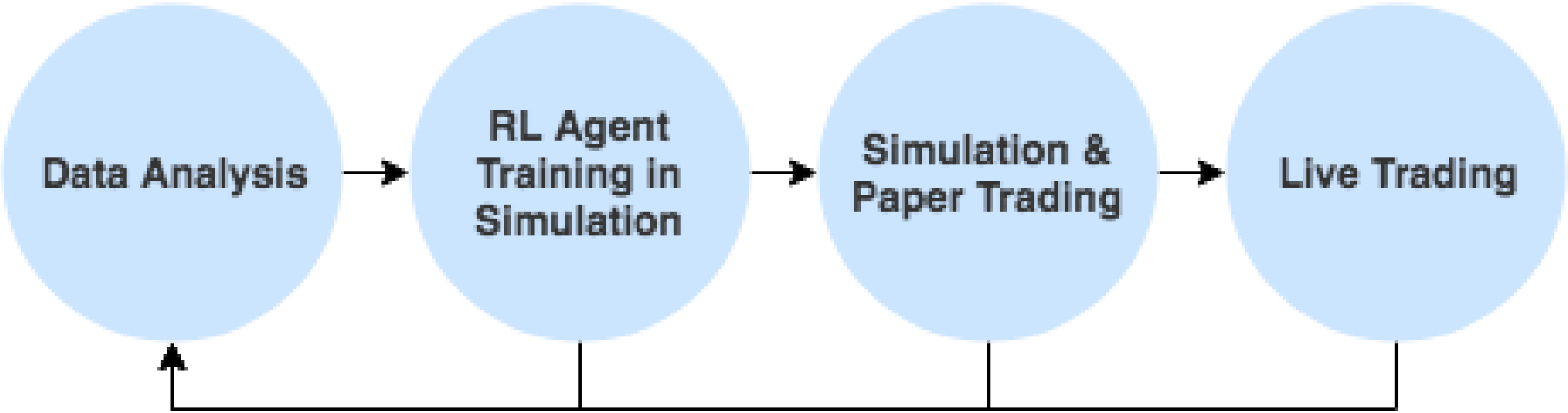
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford Book.

Reinforcement Learning Algorithms



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

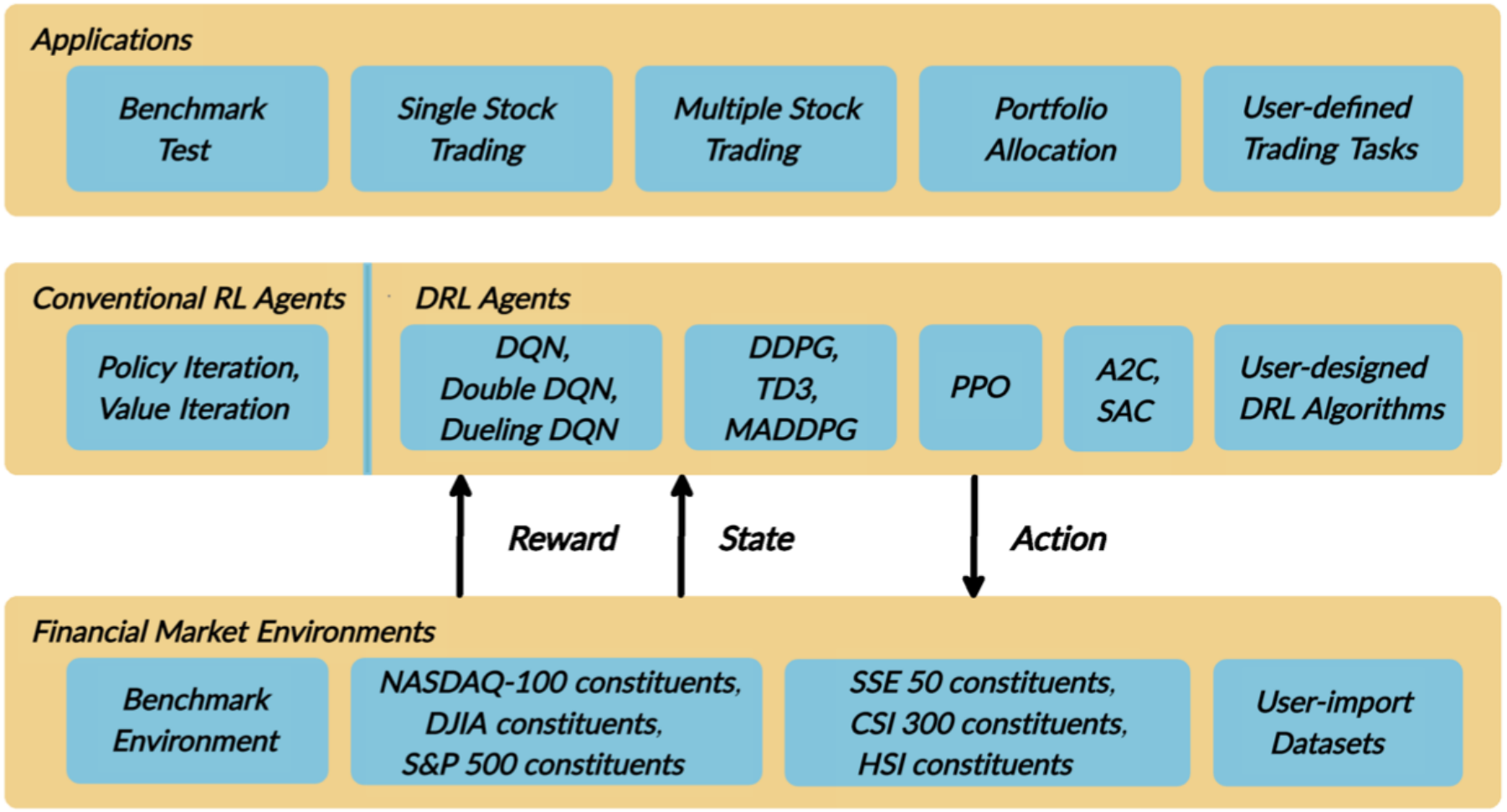
Reinforcement Learning (RL) in Trading Strategies



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

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

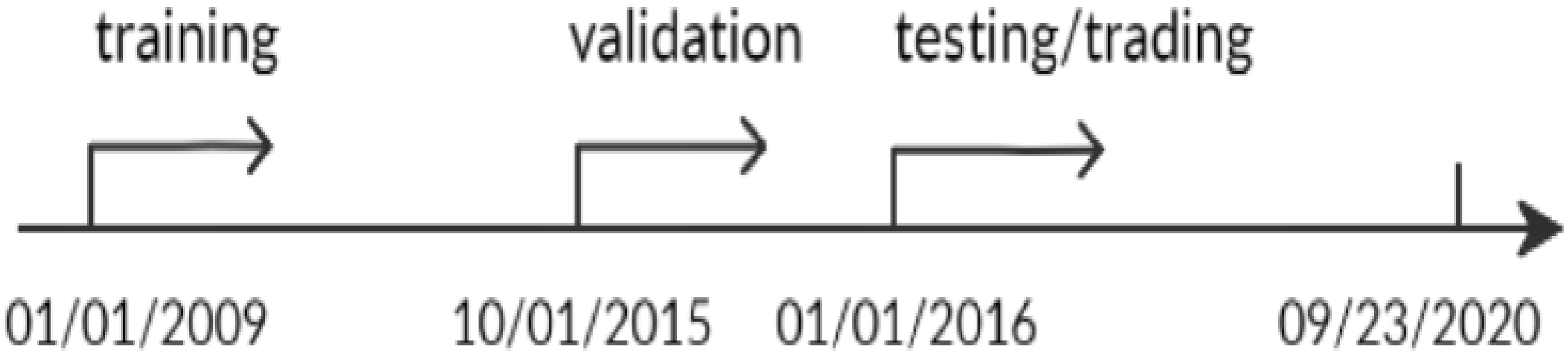
Source: Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang (2020). "FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance." arXiv preprint arXiv:2011.09607 (2020).

FinRL:

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

Evaluation of Trading Performance

Training-Validation-Testing Flow



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).

Reinforcement Learning (RL)

FinRL

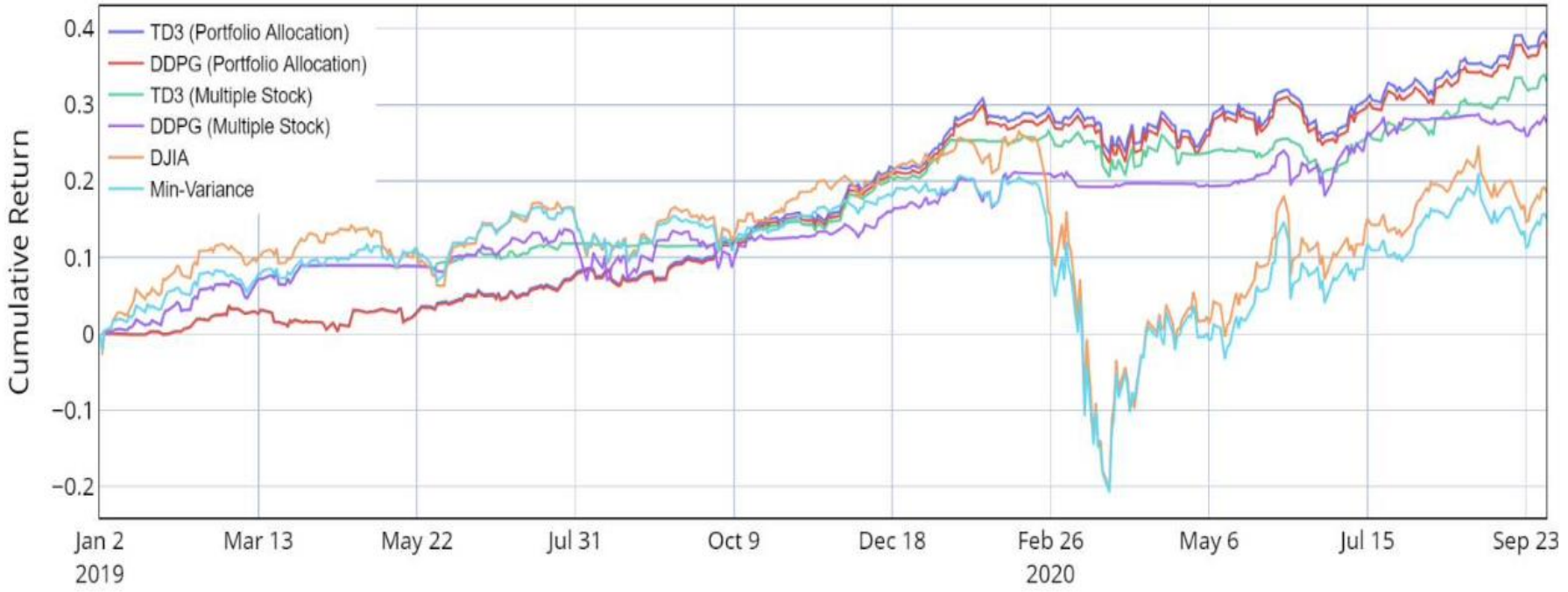
Performance of single stock trading

using Proximal policy optimization (PPO) in the FinRL library

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).

Reinforcement Learning (RL)
FinRL

Performance of multiple stock trading and portfolio allocation using the FinRL library



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).

Reinforcement Learning (RL)

FinRL

Performance of single stock trading using Proximal policy optimization (PPO) in the FinRL library

2019/01/01-2020/09/23	SPY	QQQ	GOOGL	AMZN	AAPL	MSFT	S&P 500
Initial value	100,000	100,000	100,000	100,000	100,000	100,000	100,000
Final value	127,044	163,647	174,825	192,031	173,063	172,797	133,402
Annualized return	14.89%	32.33%	37.40%	44.94%	36.88%	36.49%	17.81%
Annualized Std	9.63%	27.51%	33.41%	29.62%	25.84%	33.41%	27.00%
Sharpe ratio	1.49	1.16	1.12	1.40	1.35	1.10	0.74
Max drawdown	20.93%	28.26%	27.76%	21.13%	22.47%	28.11%	33.92%

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).

Reinforcement Learning (RL)

FinRL

Performance of multiple stock trading and portfolio allocation

over the DJIA constituents stocks using the FinRL library

2019/01/01-2020/09/23	TD3	DDPG	Min-Var.	DJIA
Initial value	1,000,000	1,000,000	1,000,000	1,000,000
Final value	1,403,337; 1,381,120	1,396,607; 1,281,120	1,171,120	1,185,260
Annualized return	21.40%; 17.61%	20.34%; 15.81%	8.38%	10.61%
Annualized Std	14.60%; 17.01%	15.89%; 16.60%	26.21%	28.63%
Sharpe ratio	1.38; 1.03	1.28; 0.98	0.44	0.48
Max drawdown	11.52% 12.78%	13.72%; 13.68%	34.34%	37.01%

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).

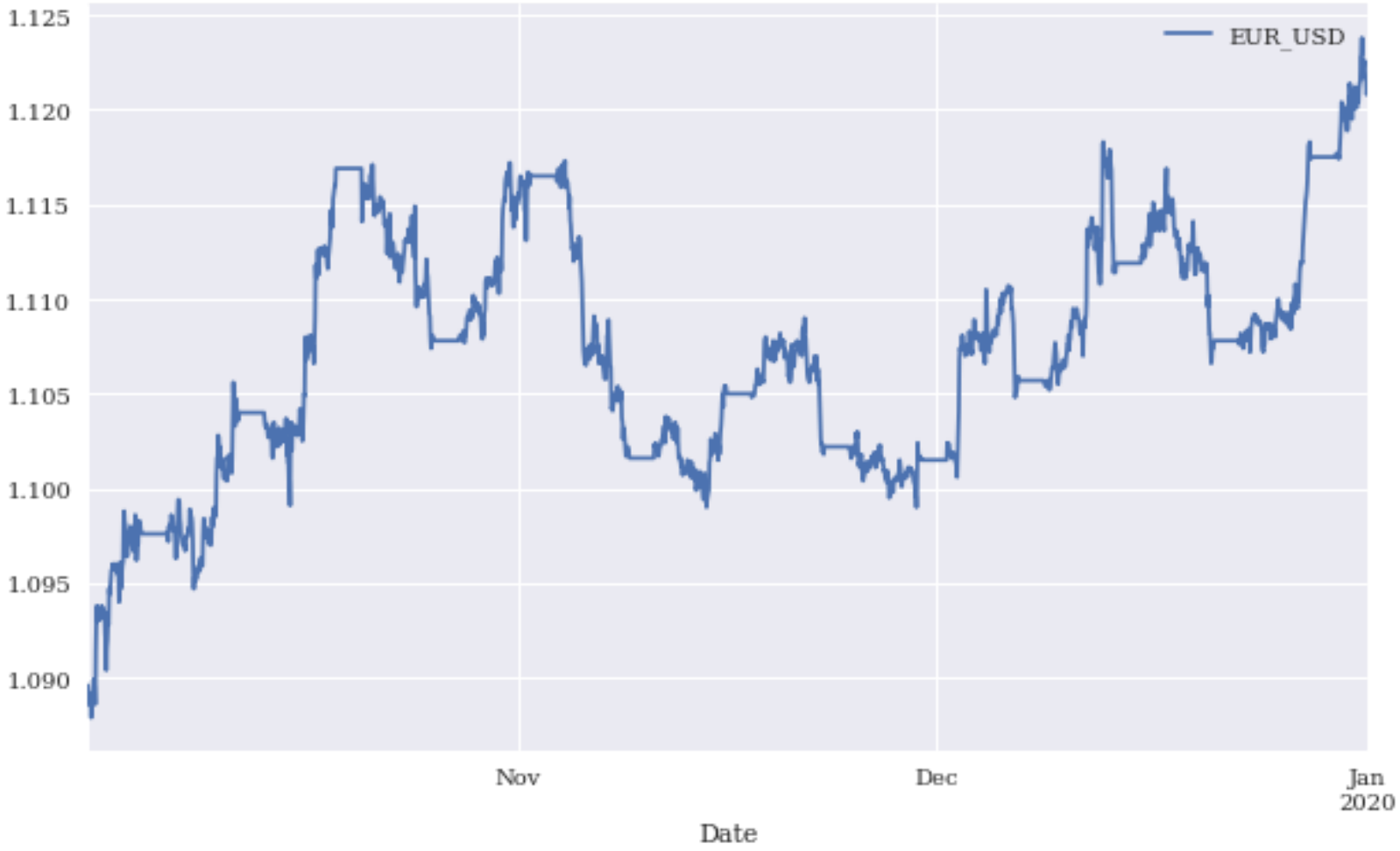
Deep Reinforcement Learning Library

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

```
import os
import numpy as np
import pandas as pd
from pylab import plt, mpl
plt.style.use('seaborn')
mpl.rcParams['savefig.dpi'] = 300
mpl.rcParams['font.family'] = 'serif'
pd.set_option('precision', 4)
np.set_printoptions(suppress=True, precision=4)
os.environ['PYTHONHASHSEED'] = '0'
```

```
url = 'http://hilpisch.com/aiif_eikon_id_eur_usd.csv'
symbol = 'EUR_USD'
raw = pd.read_csv(url, index_col=0, parse_dates=True)
raw.head()
```

Mid-closing prices for EUR/USD (intraday)



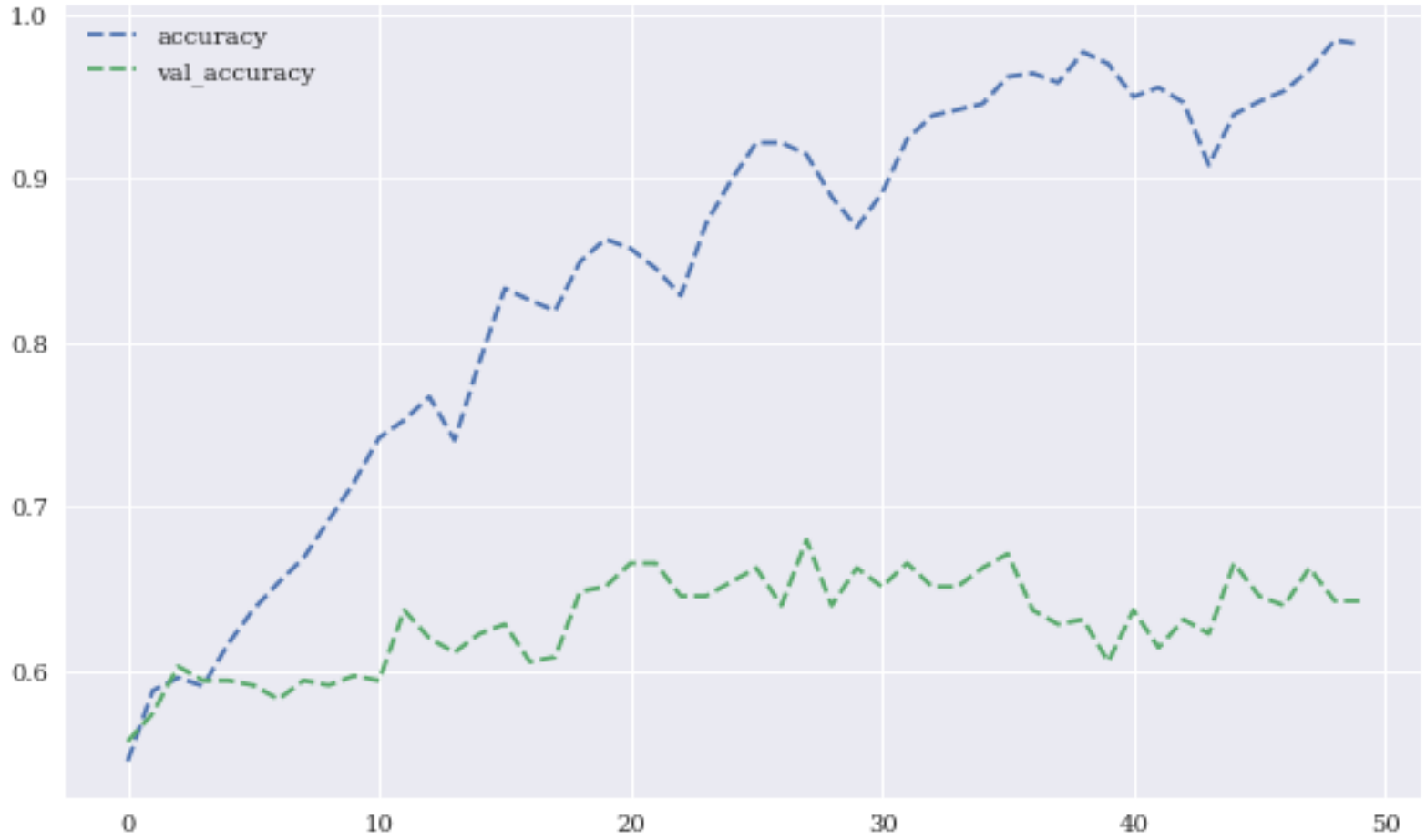
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

```
optimizer = Adam(lr=0.001)

def create_model(hl=1, hu=128, optimizer=optimizer):
    model = Sequential()
    model.add(Dense(hu, input_dim=len(cols),
                    activation='relu'))
    for _ in range(hl):
        model.add(Dense(hu, activation='relu'))
    model.add(Dense(1, activation='sigmoid'))
    model.compile(loss='binary_crossentropy',
                  optimizer=optimizer,
                  metrics=['accuracy'])
    return model

set_seeds()
model = create_model(hl=1, hu=128)
model.summary()
```

Training and validation accuracy values

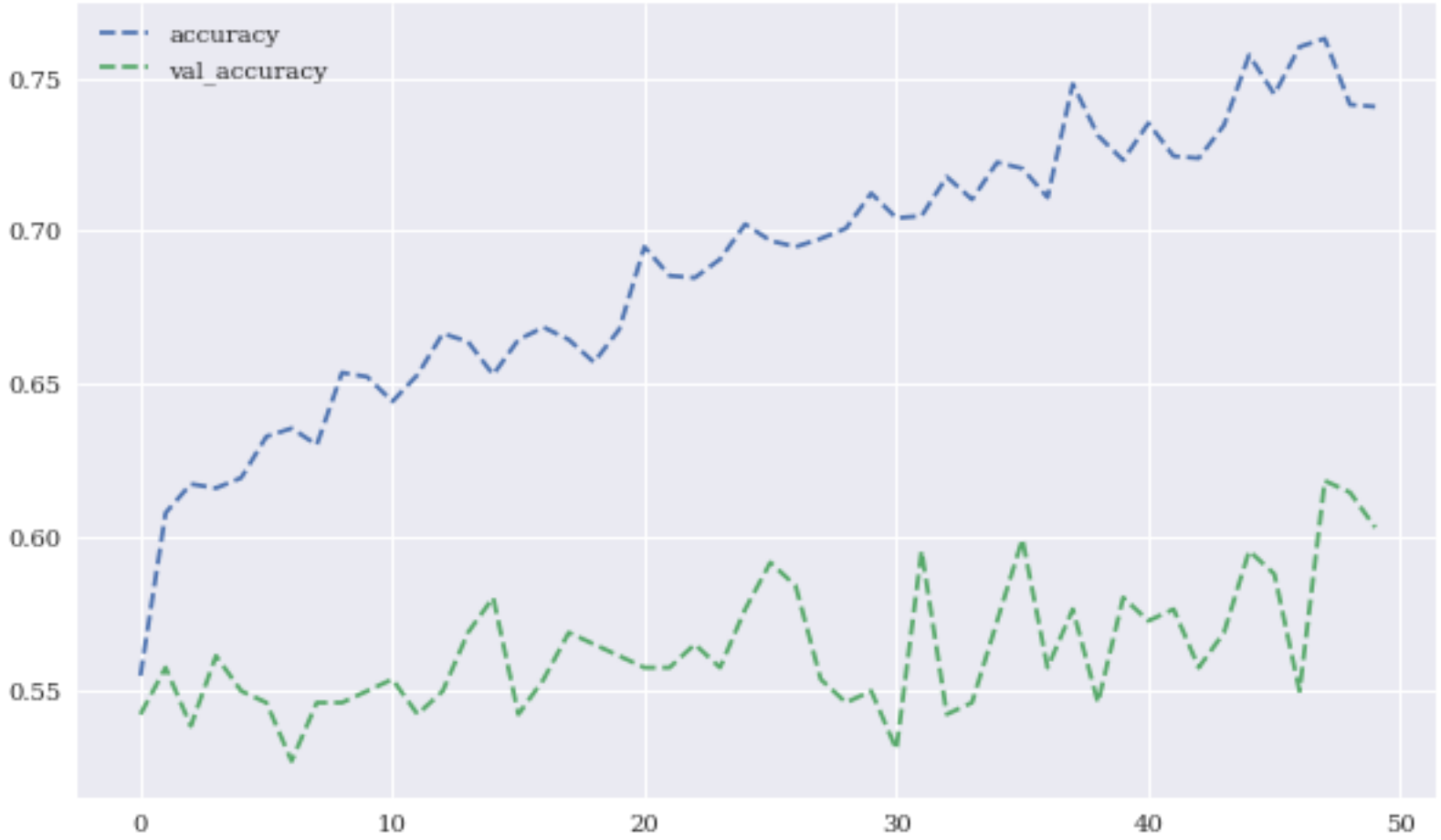
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Training and validation accuracy values (normalized features data)



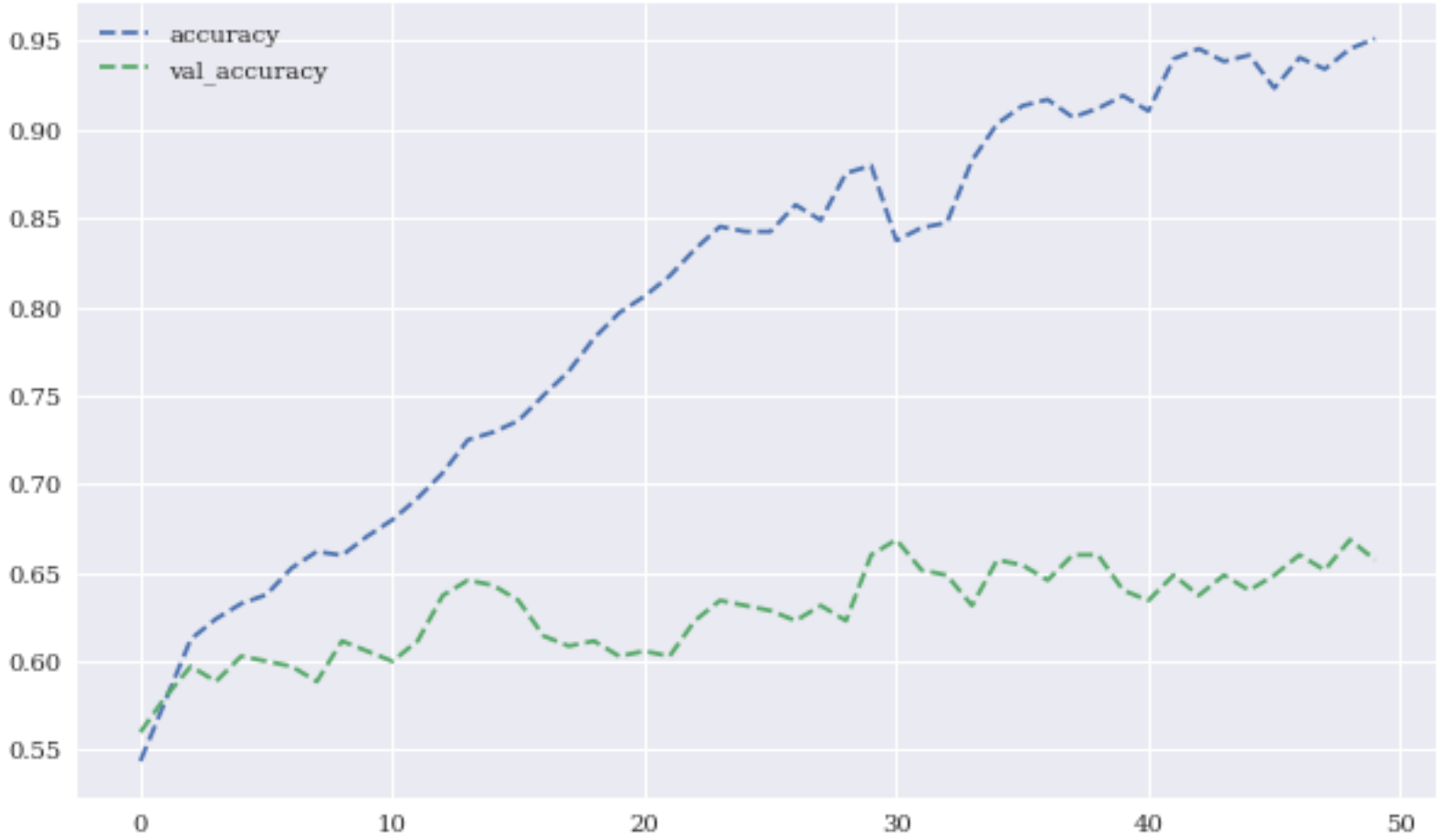
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Training and validation accuracy values (with dropout)



Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Training and validation accuracy values (with regularization)



Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

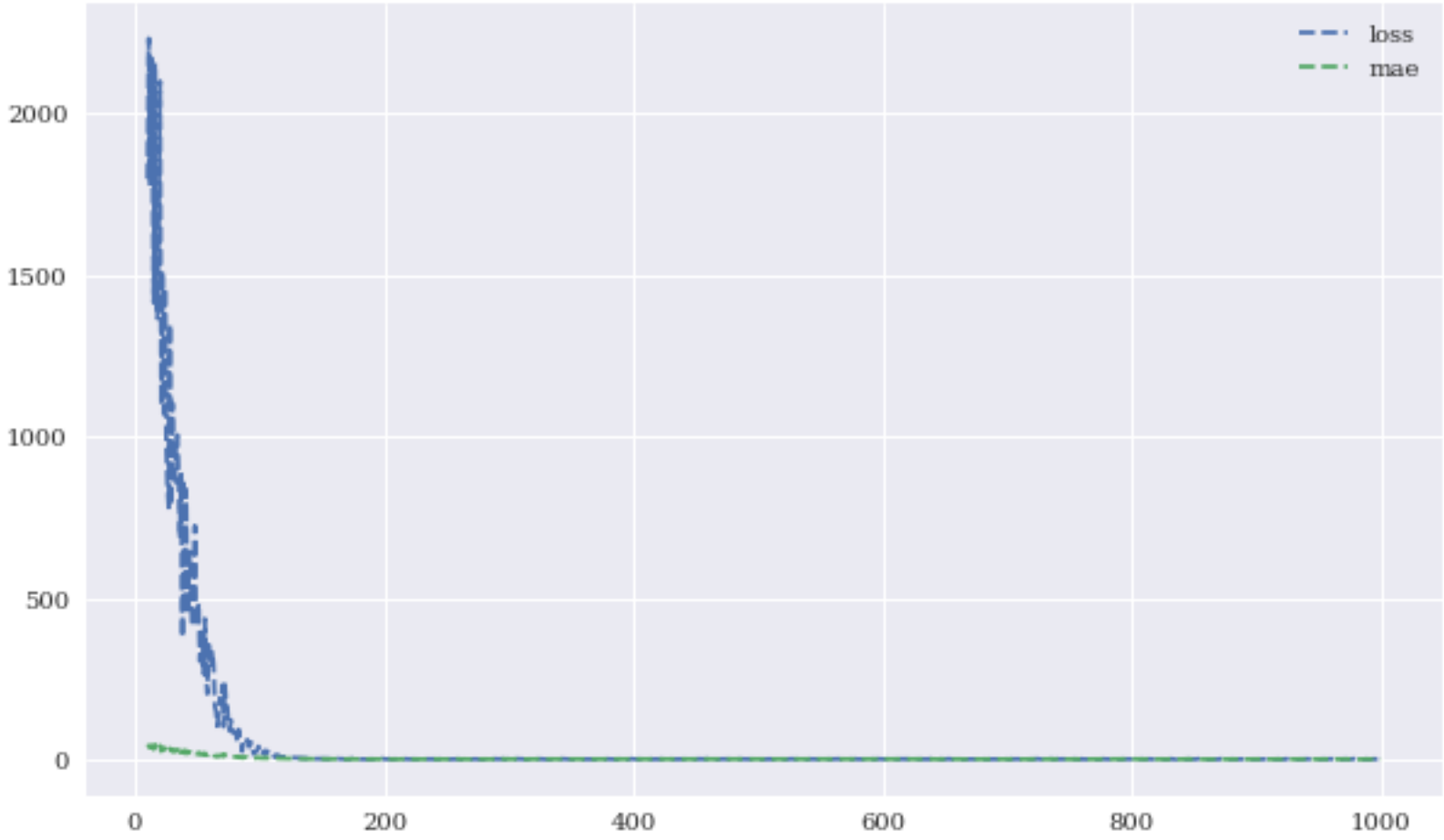
Training and validation accuracy values (with dropout and regularization)

Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

```
from keras.models import Sequential
from keras.layers import SimpleRNN, LSTM, Dense

model = Sequential()
model.add(SimpleRNN(100, activation='relu',
          input_shape=(lags, 1)))
model.add(Dense(1, activation='linear'))
model.compile(optimizer='adagrad', loss='mse',
              metrics=['mae'])

model.summary()
```

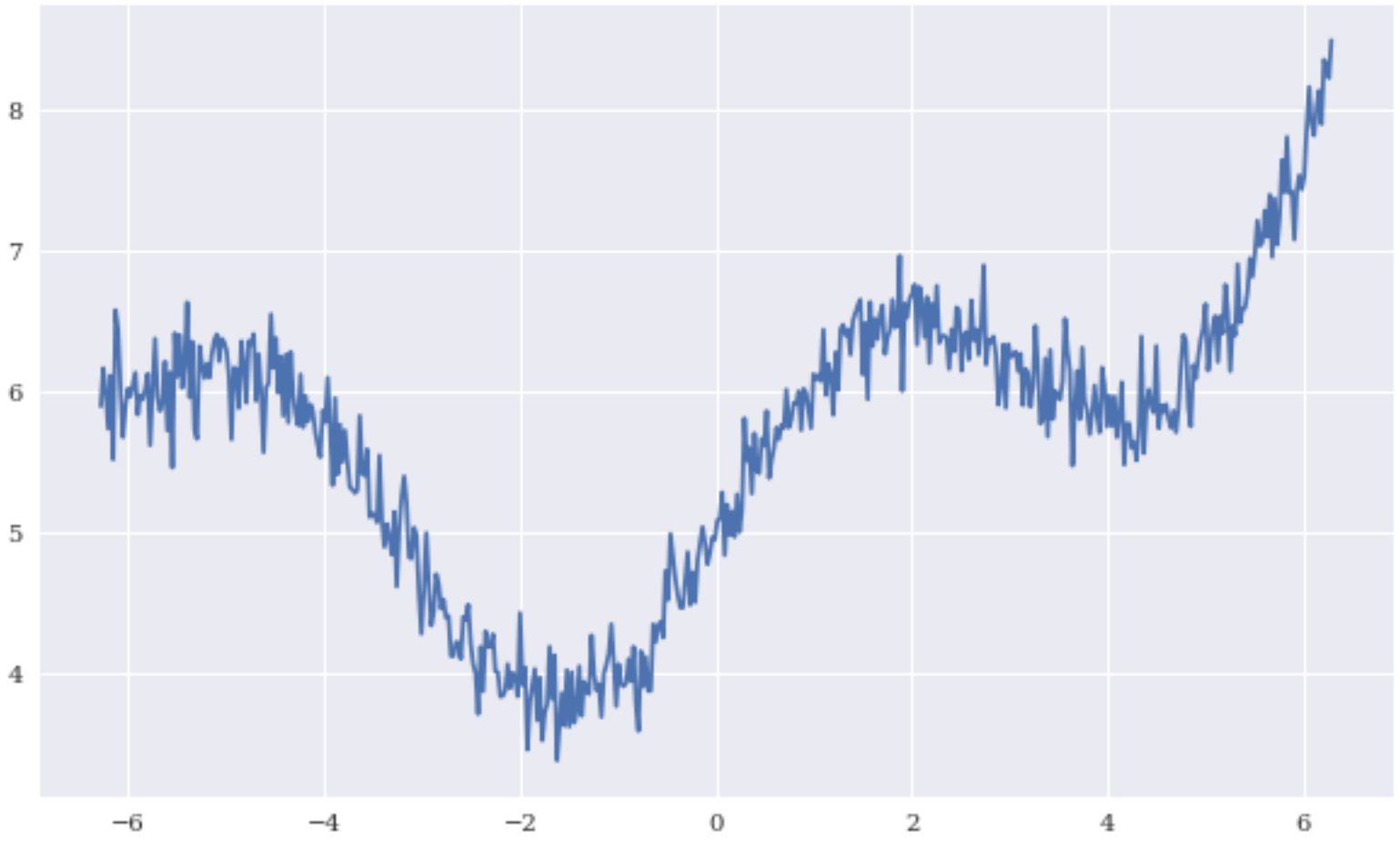
```
model.fit(g, epochs=1000, steps_per_epoch=5, verbose=False)
```

Performance metrics during RNN training



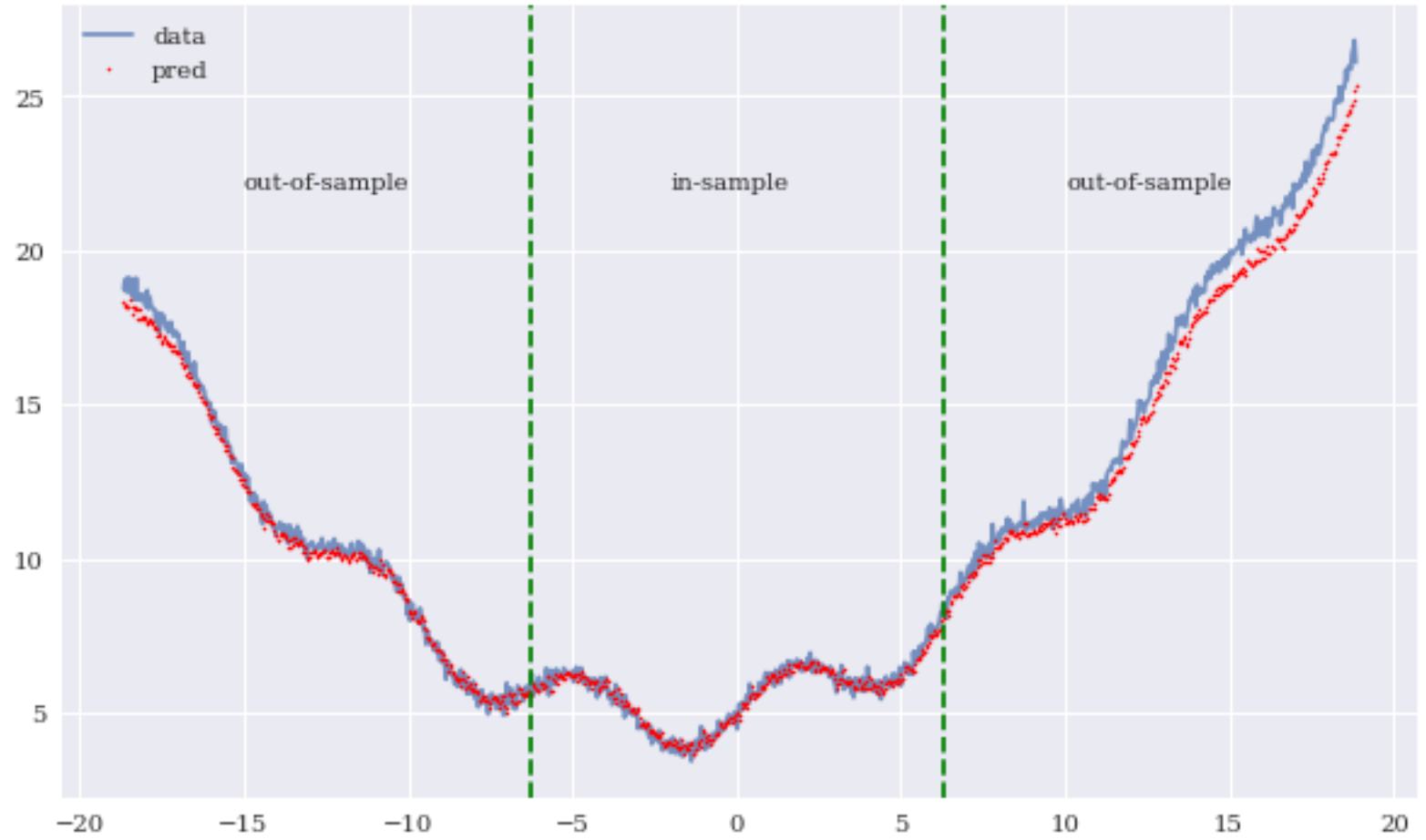
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Sample sequence data



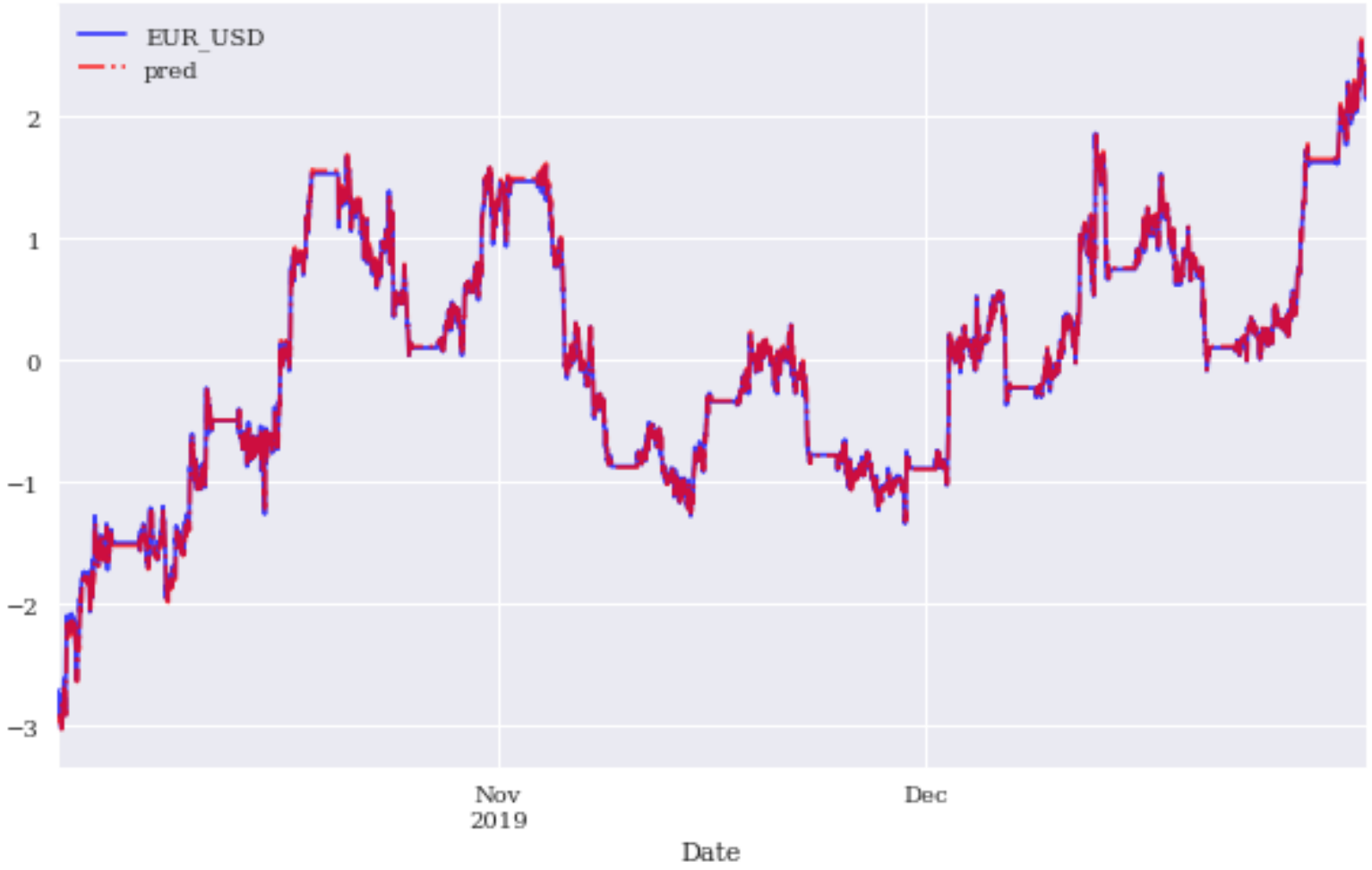
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

in-sample and out-of-sample predictions of the RNN



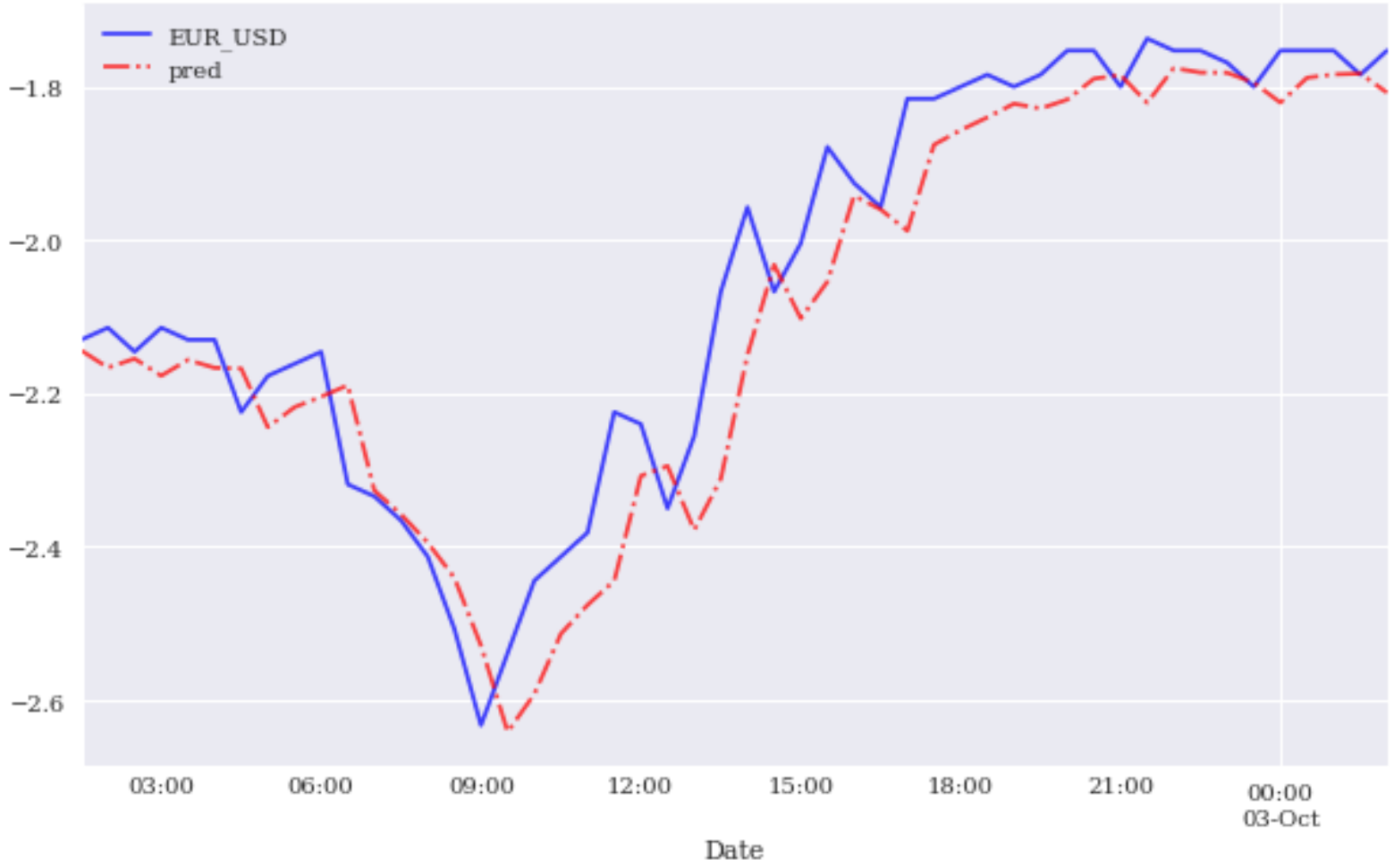
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

In-sample prediction for financial price series by the RNN (whole data set)



Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

In-sample prediction for financial price series by the RNN (data sub-set)



Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

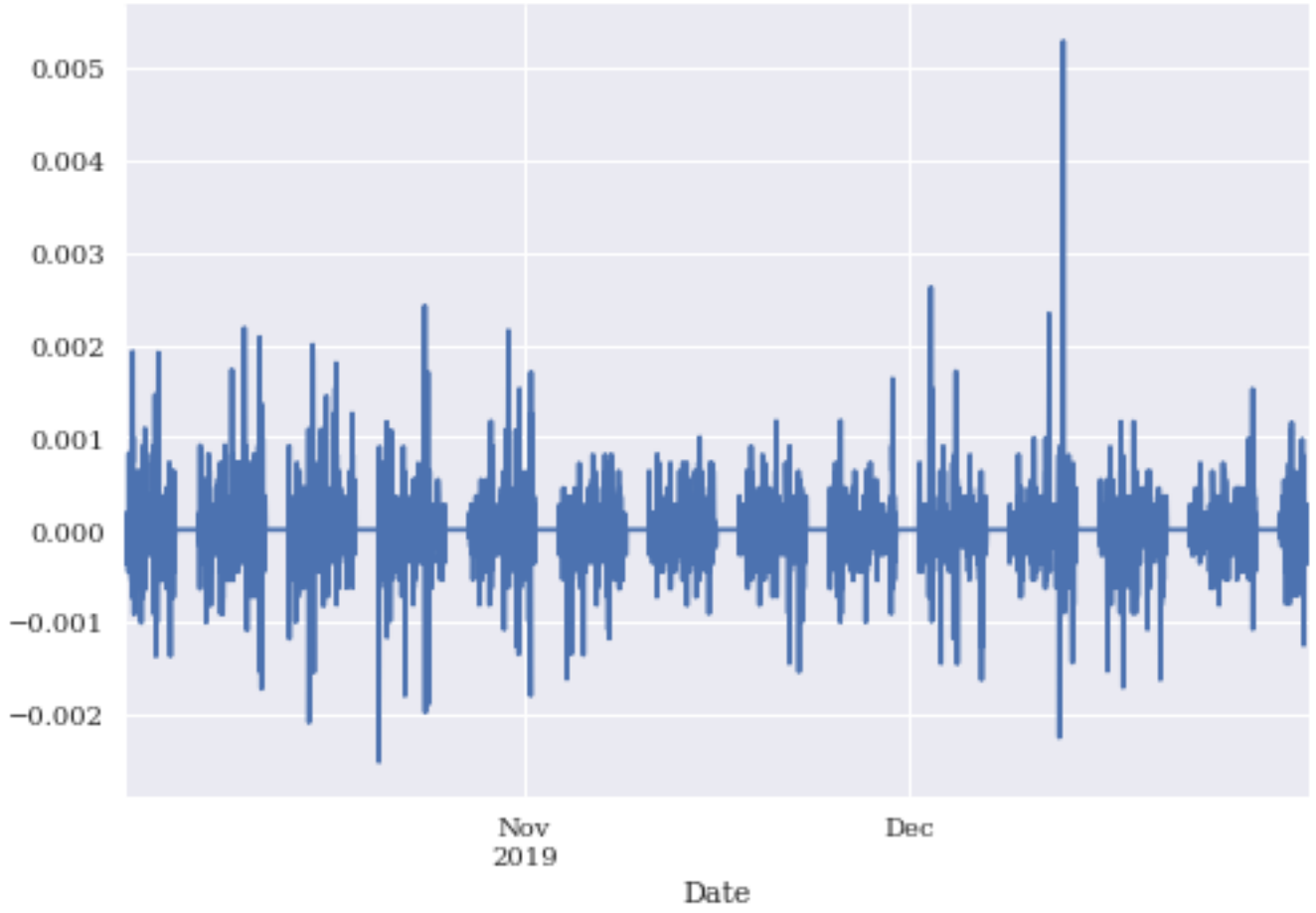
Financial Price Series

```
data = generate_data()
data.plot()
```

Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Financial Return Series

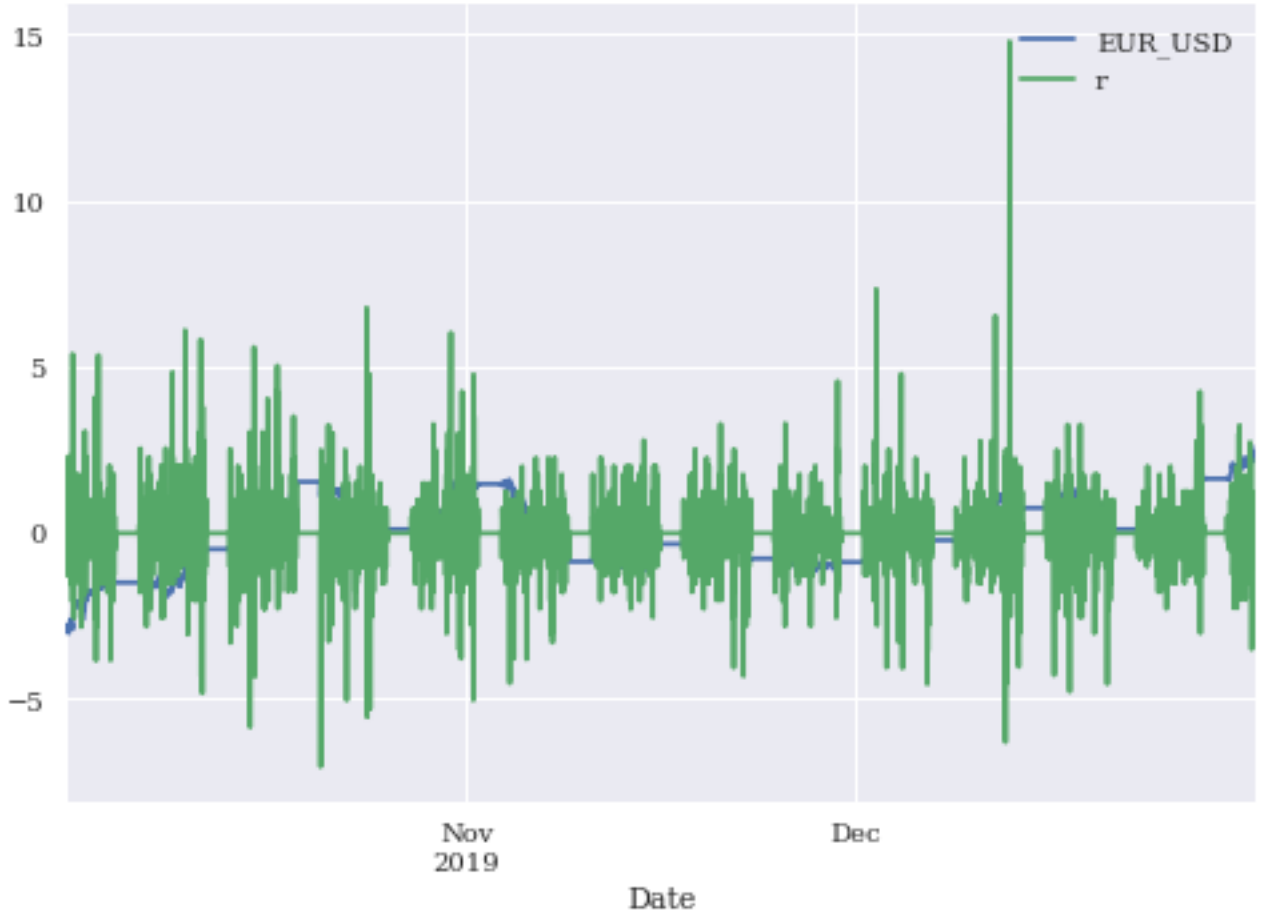
```
data['r'] = np.log(data / data.shift(1))
data['r'].plot()
```



Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

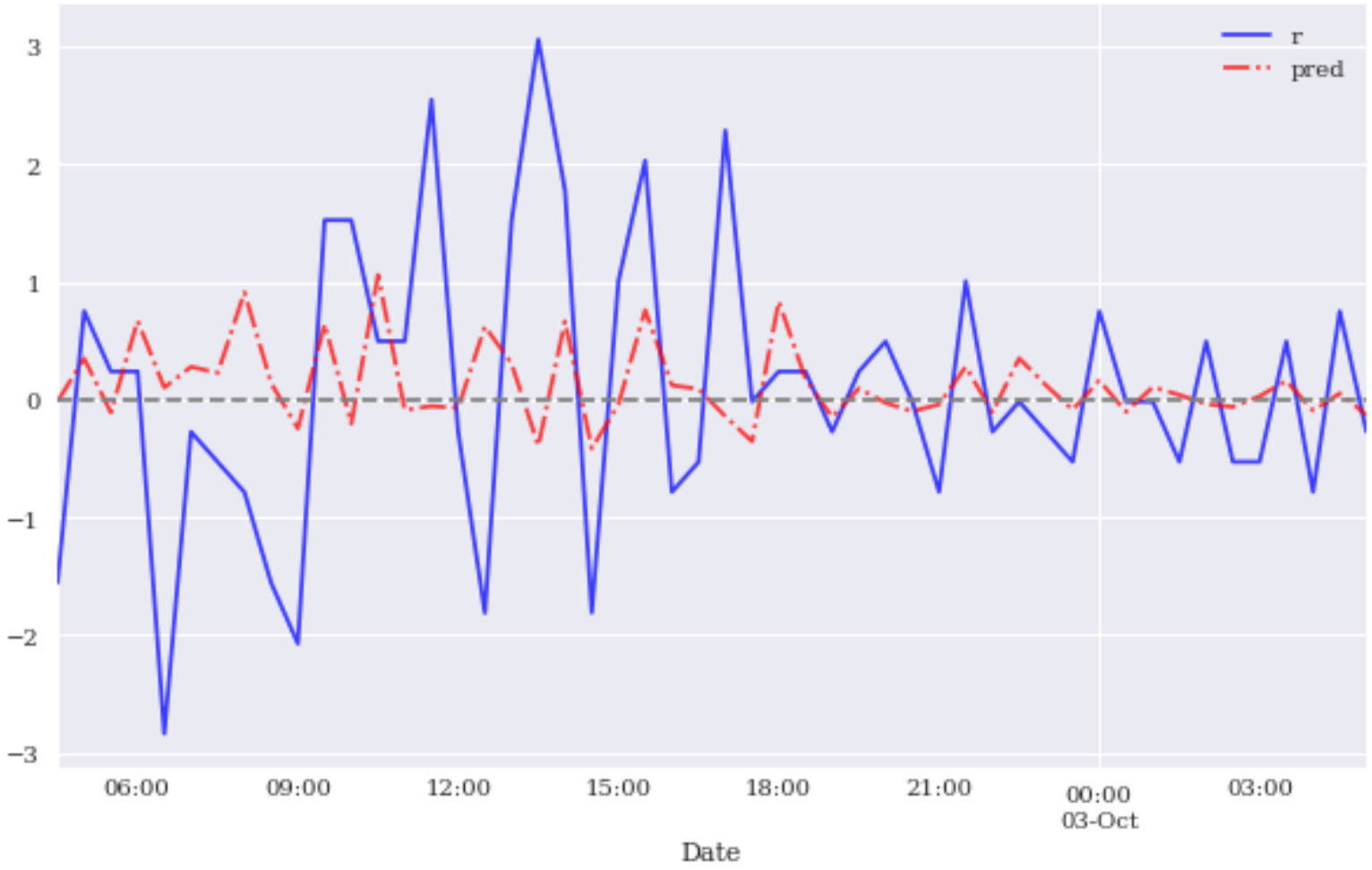
Financial Price and Return Normalization Series

```
data.dropna(inplace=True)
data = (data - data.mean()) / data.std()
data.plot()
```



Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

In-sample prediction for financial return series by the RNN (data sub-set)

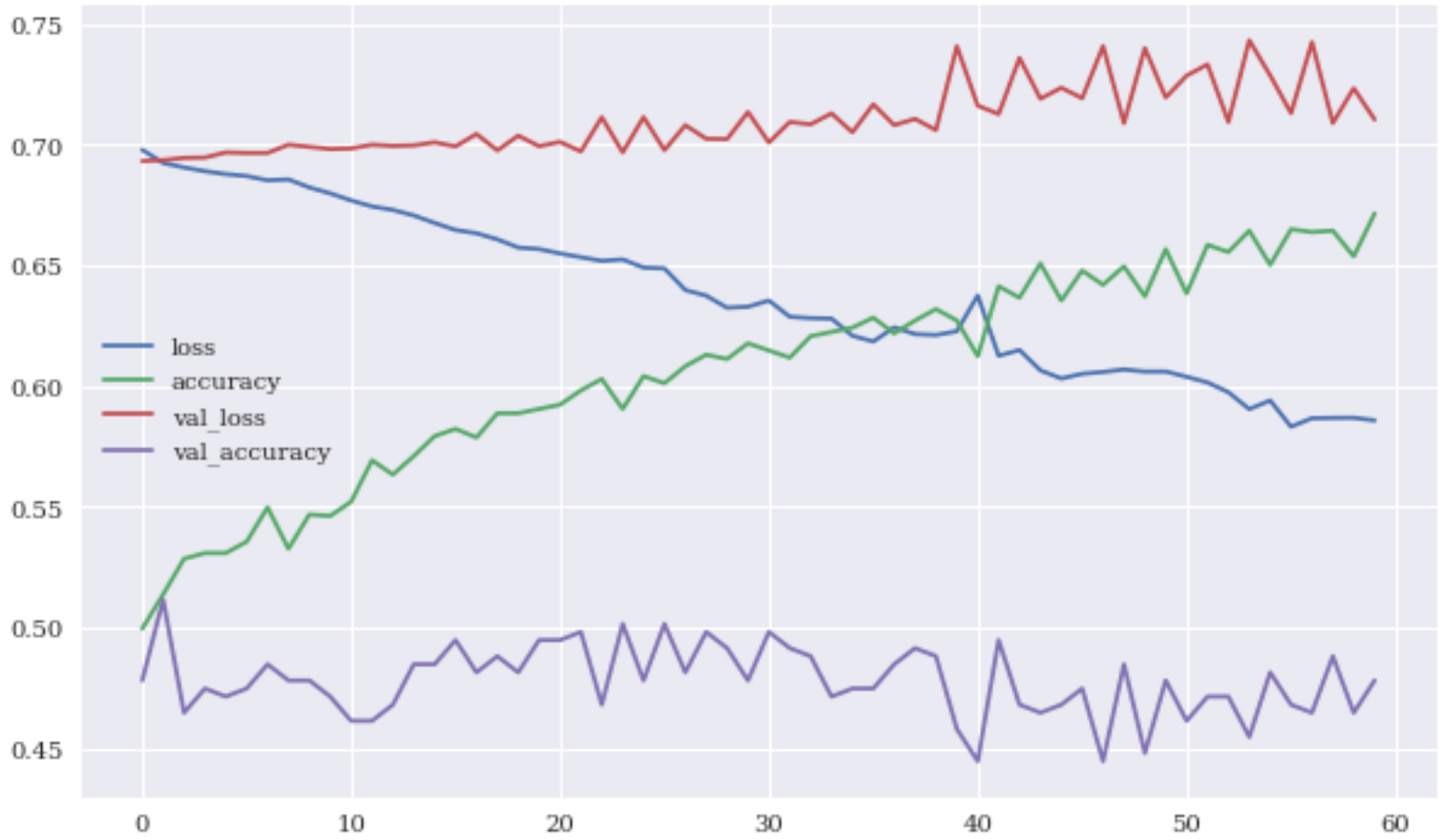


Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

```
model = Sequential()
model.add(Conv1D(filters=96, kernel_size=5,
                 activation='relu',
                 input_shape=(len(cols), 1)))
model.add(Flatten())
model.add(Dense(10, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])

model.summary()
```

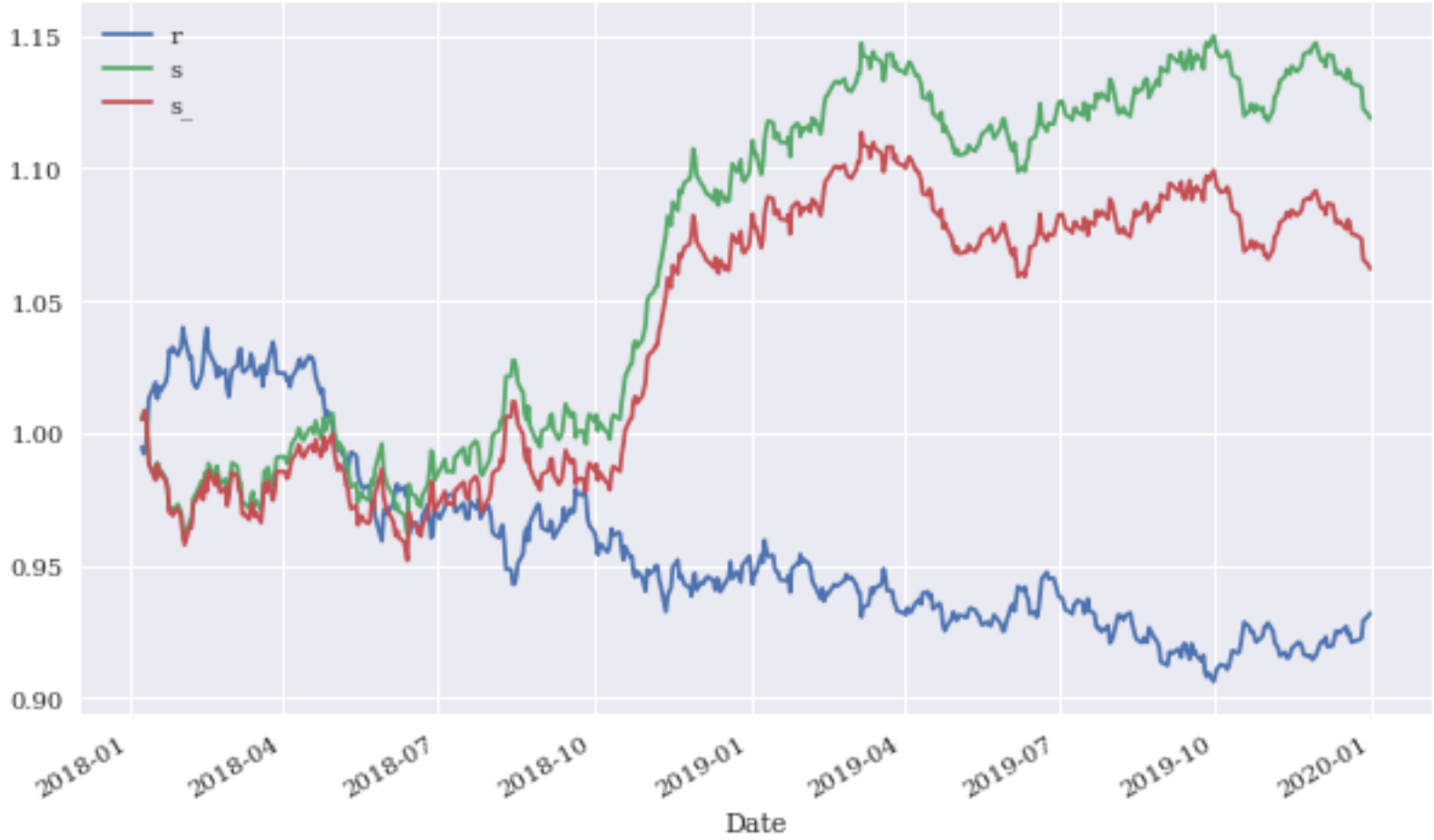
```
model.fit(np.atleast_3d(train[cols]), train['d'],
          epochs=60, batch_size=48, verbose=False,
          validation_split=0.15, shuffle=False)
```

Performance metrics for the training and validation of the CNN



Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Gross performance of passive benchmark investment and CNN strategy (before/after transaction costs)



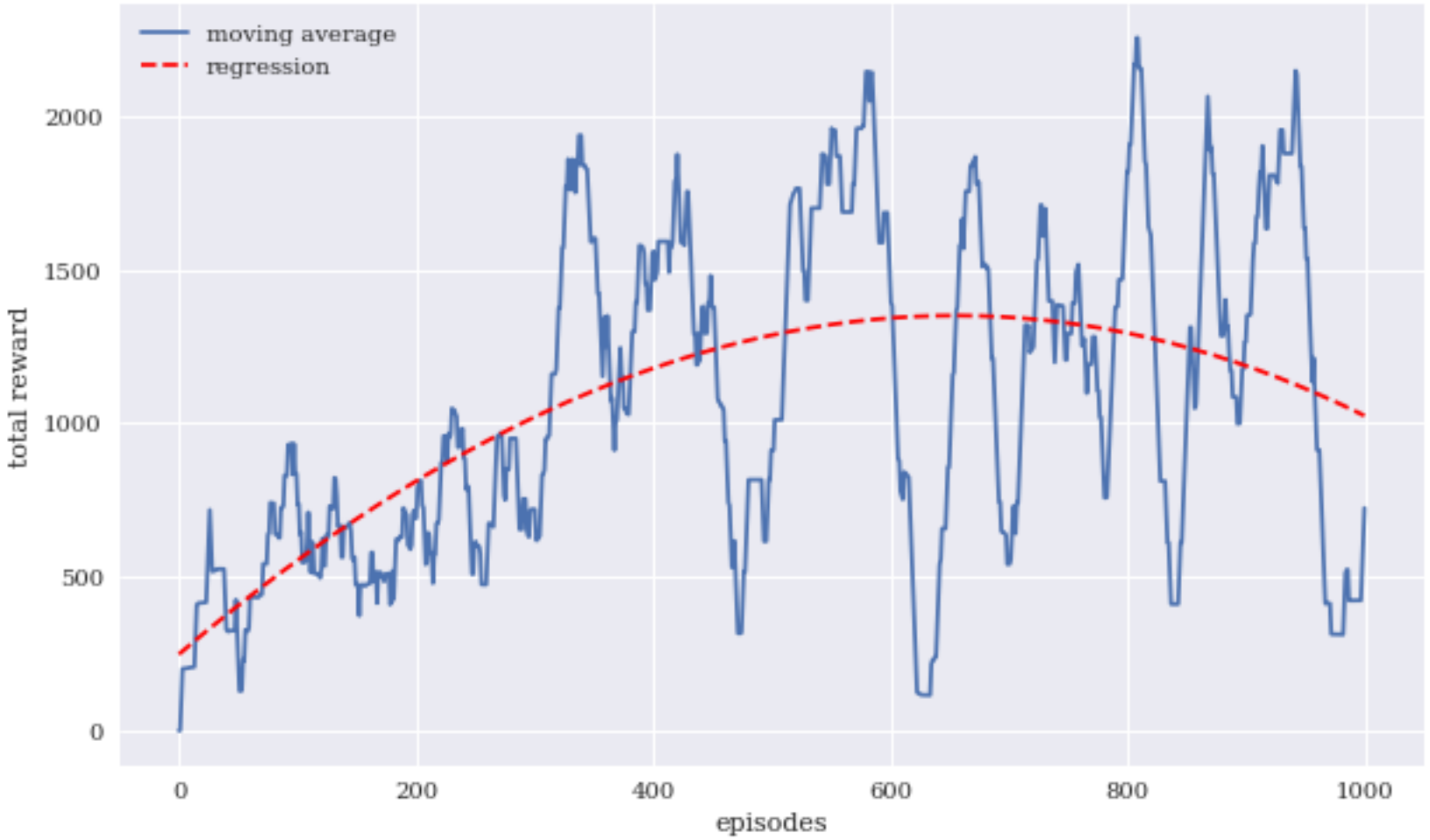
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Reinforcement Learning in Finance

- Simple Learning
- DNN Learning
- Q Learning
- Finance Environment
- Improved Finance Environment
- Improved Financial QL Agent

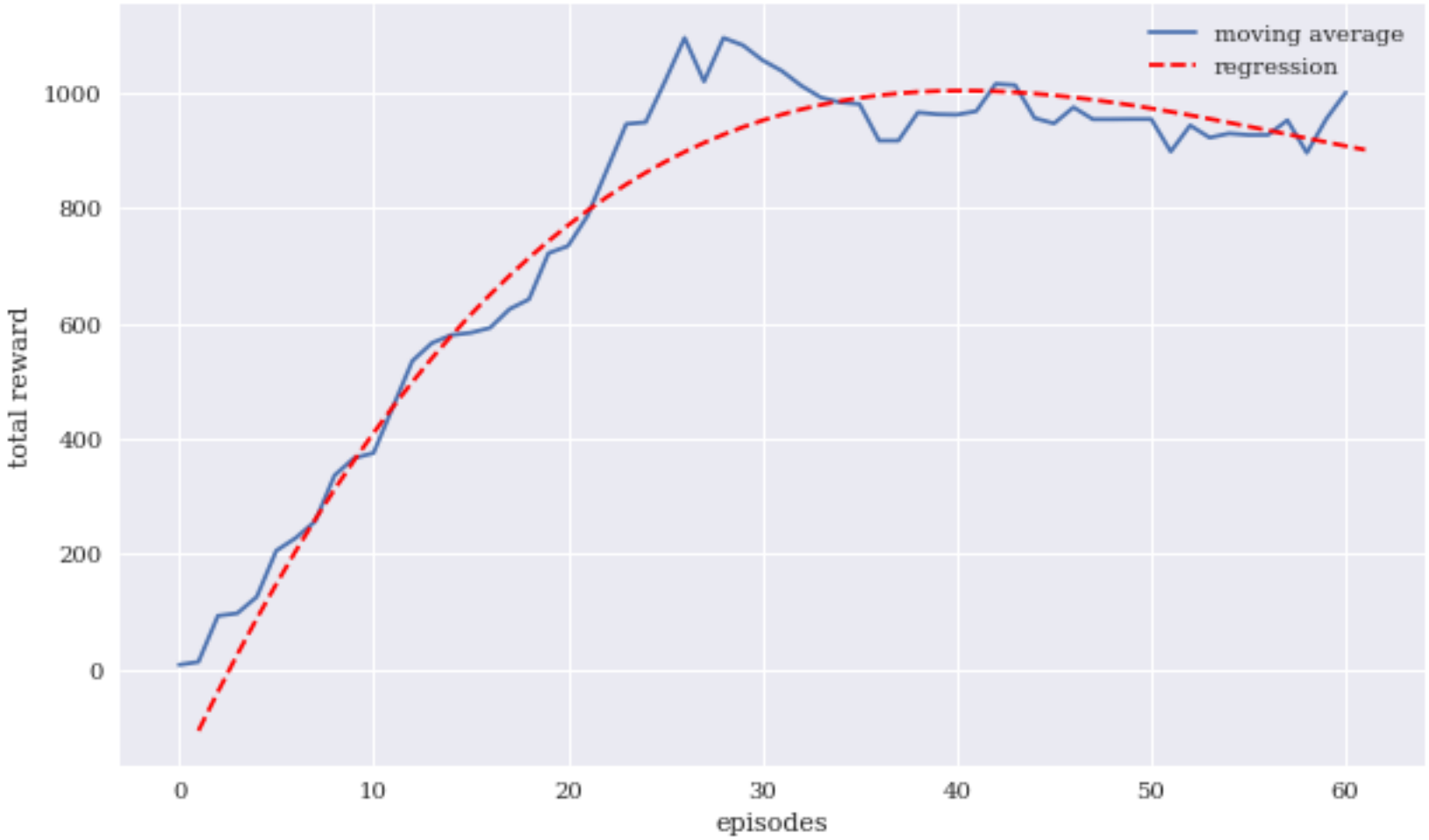
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Average total rewards of DQLAgent for CartPole



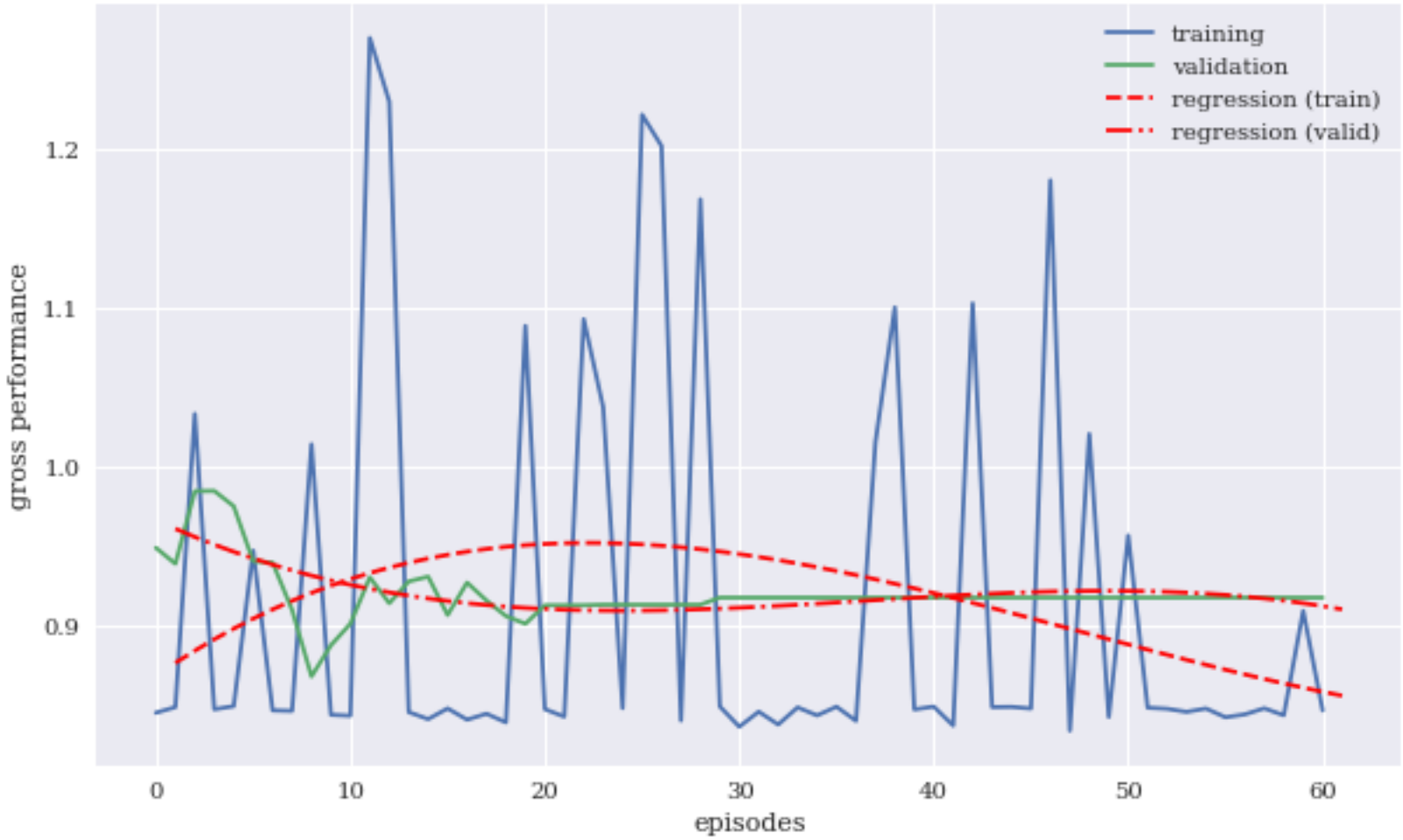
Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Average total rewards of DQLAgent for Finance



Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Training and validation performance of the FQLAgent per episode

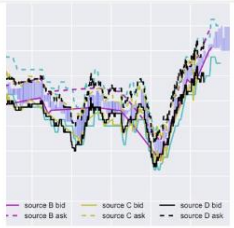


Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

Papers with Code: Time Series Analysis

Browse SoTA > Time Series > Time Series Analysis

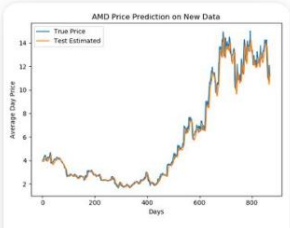
Time Series Analysis subtasks



Time Series Analysis

40 benchmarks

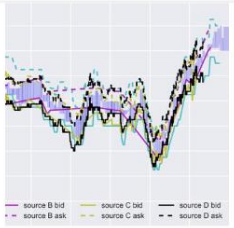
1772 papers with code



Time Series Forecasting

32 benchmarks

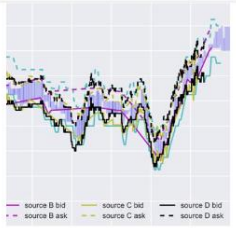
255 papers with code



Time Series Prediction

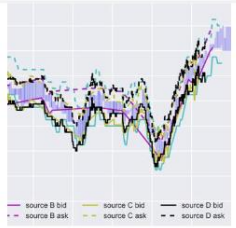
2 benchmarks

90 papers with code



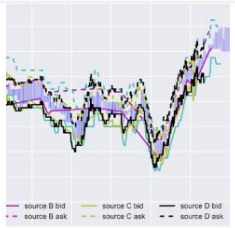
Data Compression

66 papers with code



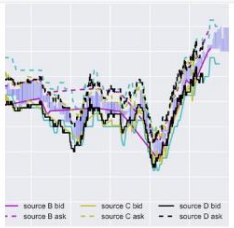
Model Optimization

62 papers with code



Time Series Anomaly Detection

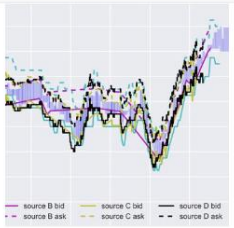
43 papers with code



Time Series Clustering

1 benchmark

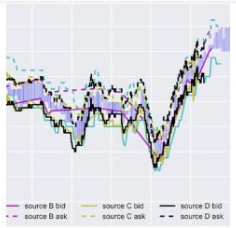
26 papers with code



Time Series Regression

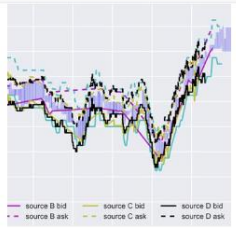
1 benchmark

22 papers with code



Time Series Averaging

3 papers with code

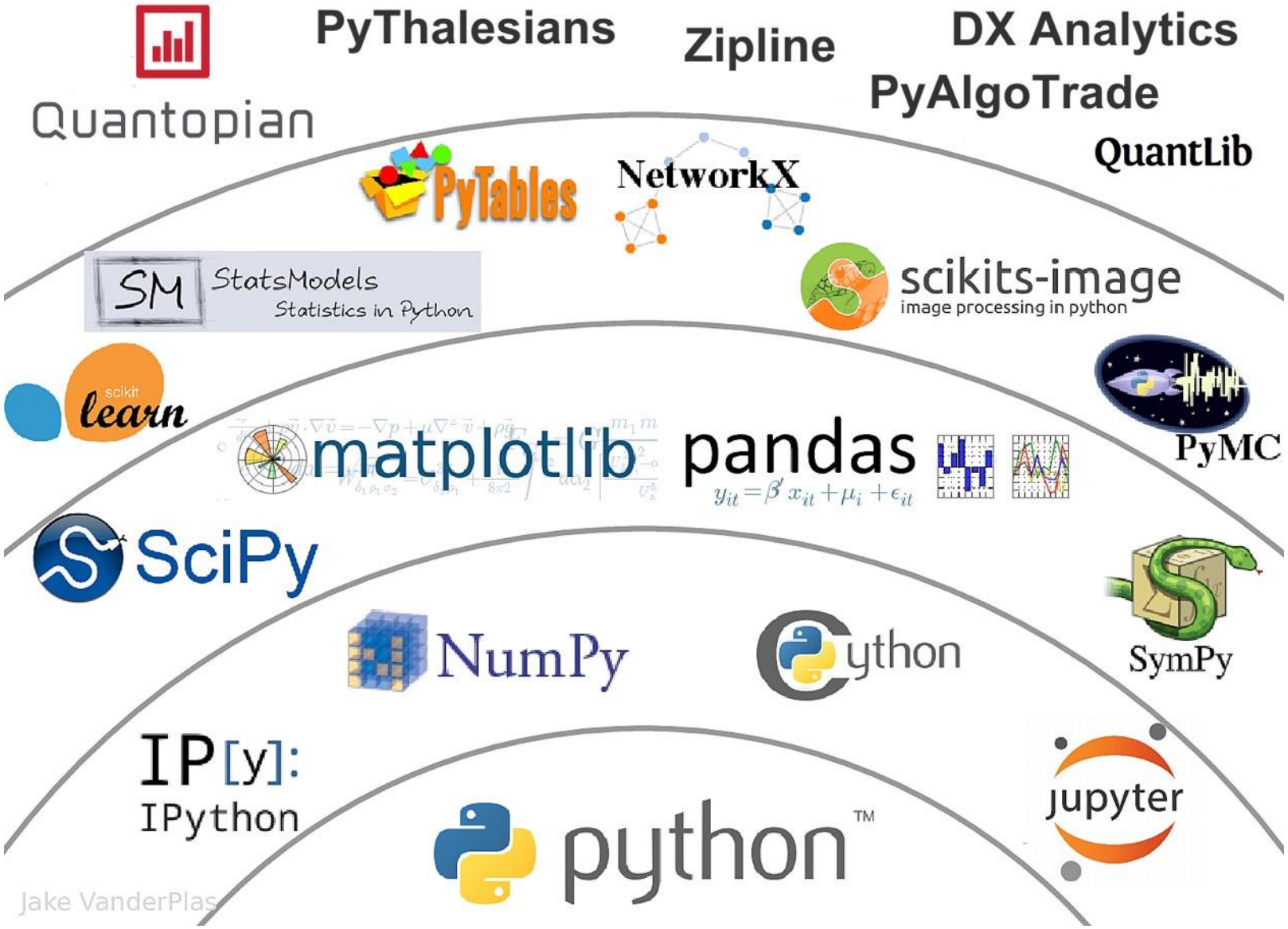


Time Series Alignment

3 papers with code

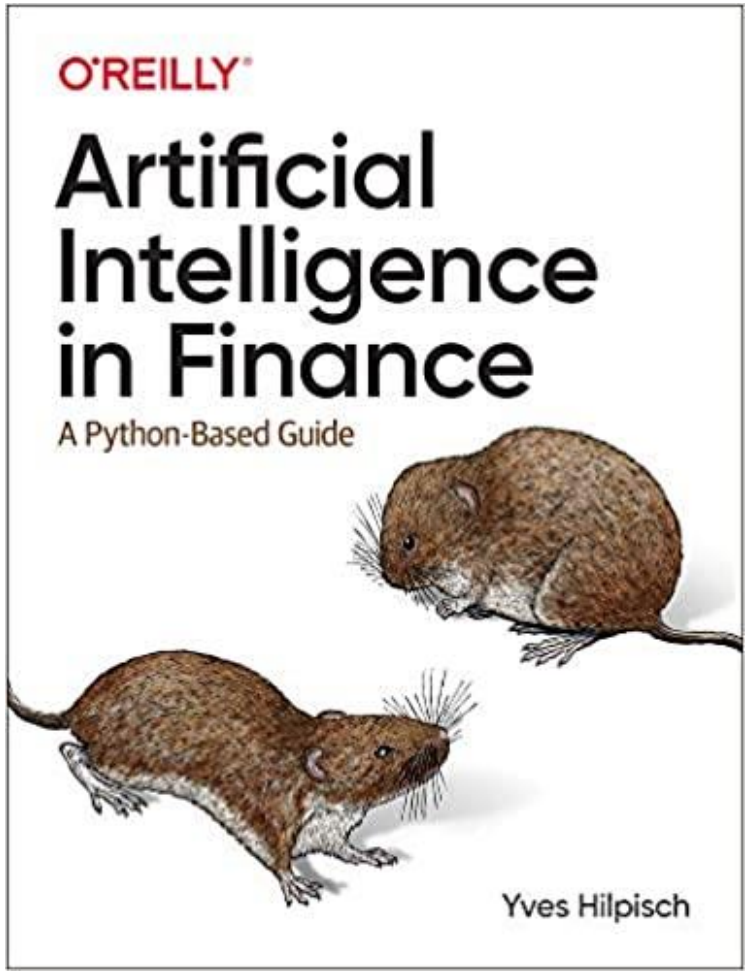
https://paperswithcode.com/area/time-series/time-series

The Quant Finance PyData Stack



Source: http://nbviewer.jupyter.org/format/slides/github/quantopian/pyfolio/blob/master/pyfolio/examples/overview_slides.ipynb#/5

Yves Hilpisch (2020),
Artificial Intelligence in Finance:
A Python-Based Guide,
O'Reilly



Source: https://www.amazon.com/Artificial-Intelligence-Finance-Python-Based-Guide/dp/1492055433

Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly

yhilpisch / aiif Public

https://github.com/yhilpisch/aiif

Notifications | Star 98 | Fork 77

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights

main | 1 branch | 0 tags | Go to file | Code

yves Code updates for TF 2.3. e334251 on Dec 8, 2020 4 commits

code	Code updates for TF 2.3.	11 months ago
.gitignore	Code updates for TF 2.3.	11 months ago
LICENSE.txt	Code updates.	11 months ago
README.md	Code updates.	11 months ago

README.md

Artificial Intelligence in Finance

About this Repository

This repository provides Python code and Jupyter Notebooks accompanying the **Artificial Intelligence in Finance** book published by O'Reilly.

O'REILLY

About

Jupyter Notebooks and code for the book Artificial Intelligence in Finance (O'Reilly) by Yves Hilpisch.

home.tpq.io/books/aiif

Readme

View license

Releases

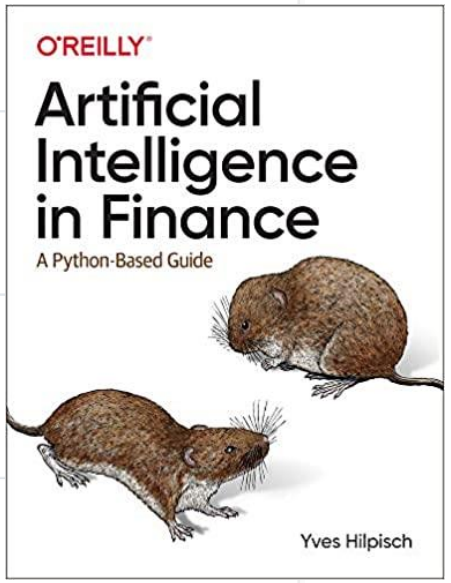
No releases published

Packages

No packages published

Languages

Jupyter Notebook 97.4% Python 2.6%



Source: https://github.com/yhilpisch/aiif

Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly

yhilpisch / aiif Public

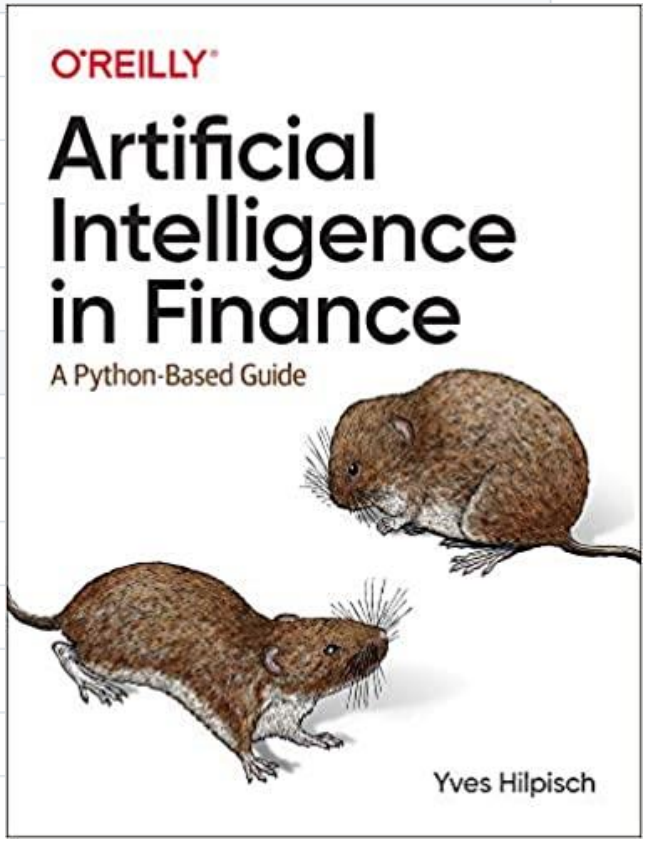
Notifications | Star 98 | Fork 77

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights

main | aiif / code / https://github.com/yhilpisch/aiif/tree/main/code | Go to file

yves Code updates for TF 2.3. e334251 on Dec 8, 2020 History

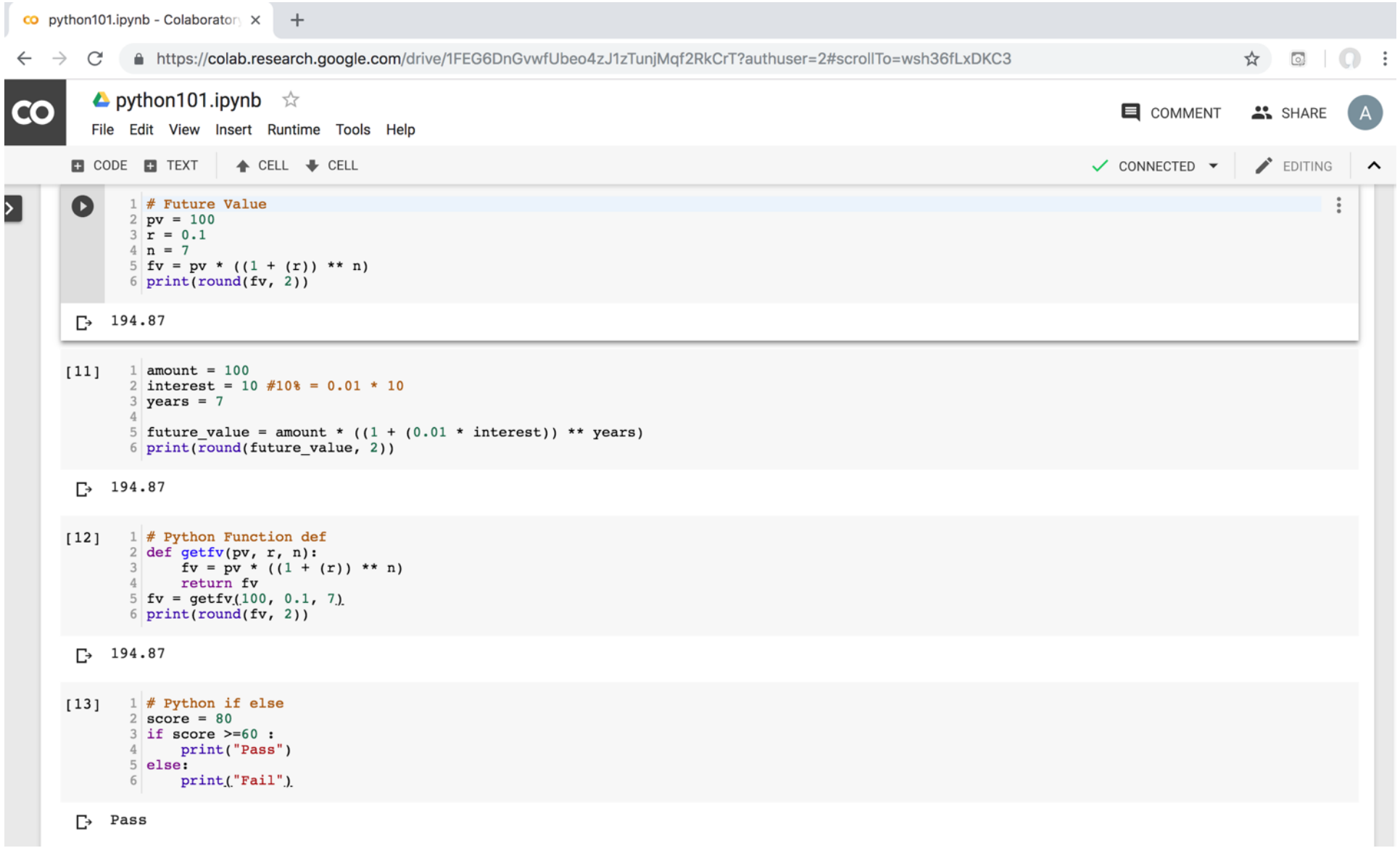
..	
oanda	Code updates for TF 2.3.
01_artificial_intelligence.ipynb	Code updates for TF 2.3.
02_superintelligence.ipynb	Code updates for TF 2.3.
03_normative_finance.ipynb	Code updates for TF 2.3.
04_data_driven_finance_a.ipynb	Initial commit.
04_data_driven_finance_b.ipynb	Initial commit.
05_machine_learning.ipynb	Code updates for TF 2.3.
06_ai_first_finance.ipynb	Code updates for TF 2.3.
07_dense_networks.ipynb	Code updates for TF 2.3.
08_recurrent_networks.ipynb	Code updates for TF 2.3.
09_reinforcement_learning_a.ipynb	Code updates.
09_reinforcement_learning_b.ipynb	Code updates for TF 2.3.



Source: https://github.com/yhilpisch/aiif/tree/main/code

Python in Google Colab (Python101)

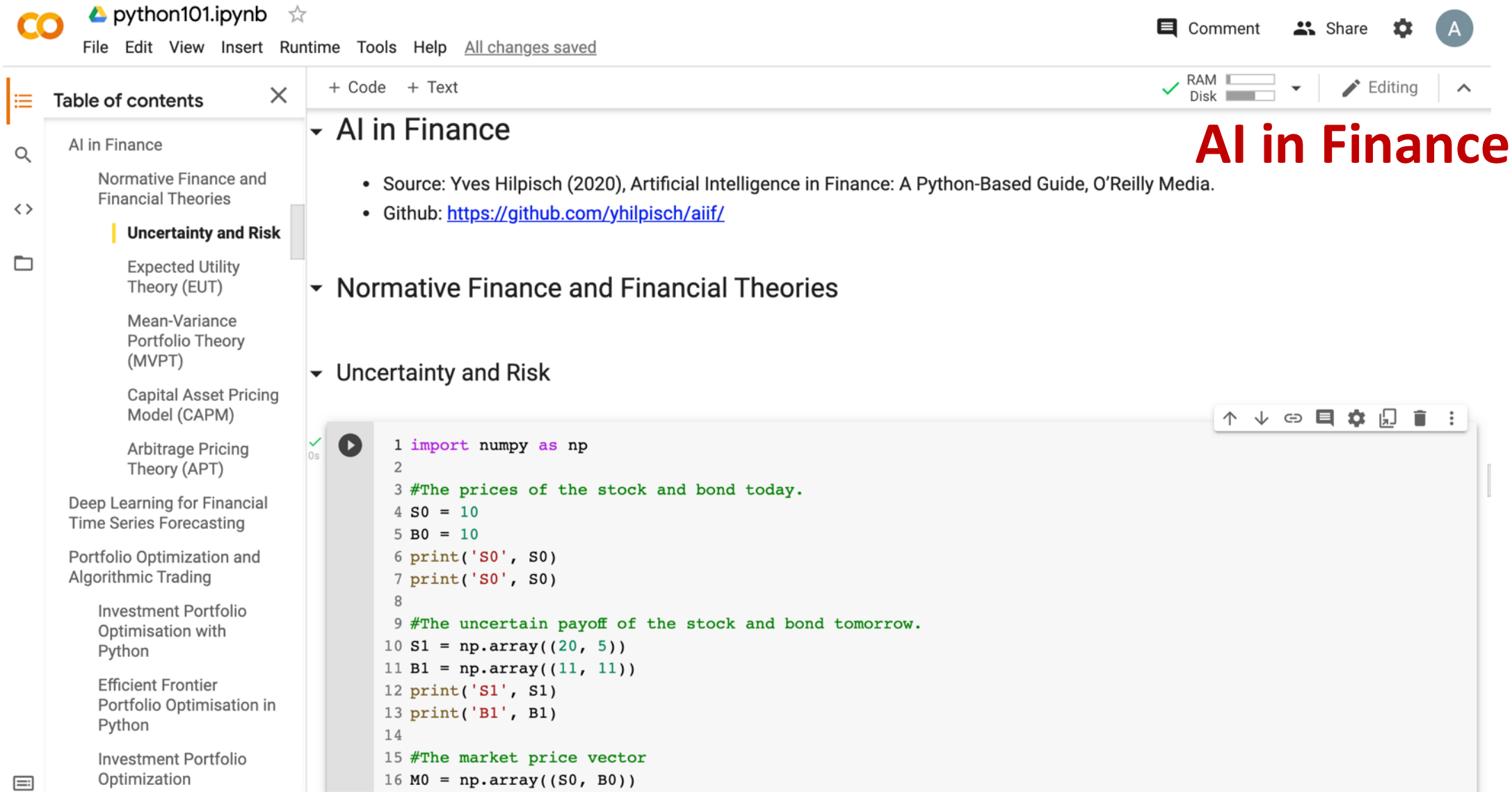
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

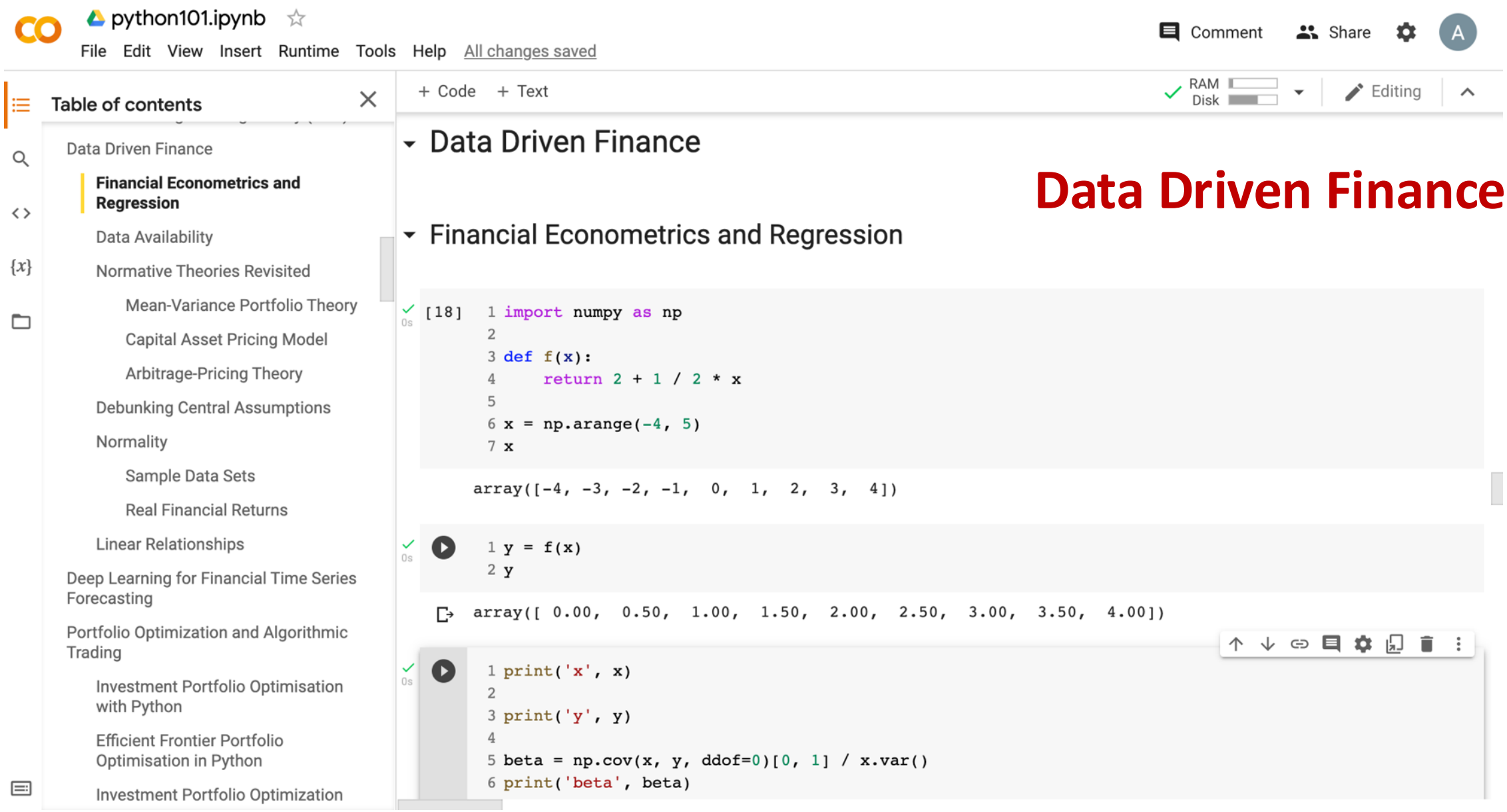
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



AI in Finance

https://tinyurl.com/aintpupython101

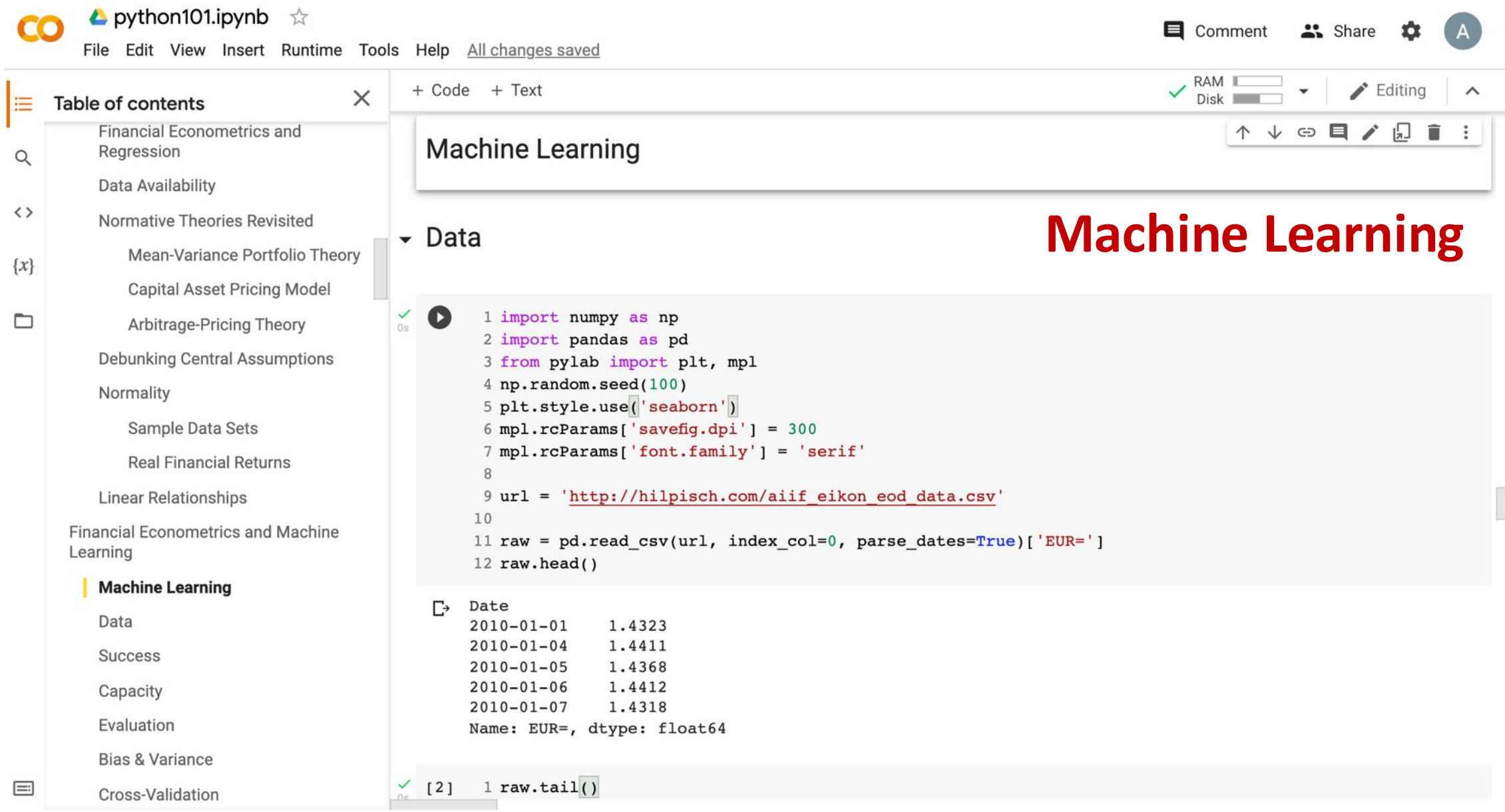
Python in Google Colab (Python101)



Data Driven Finance

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

Table of contents

- Mean-Variance Portfolio Theory
- Capital Asset Pricing Model
- Arbitrage-Pricing Theory
- Debunking Central Assumptions
- Normality
- Sample Data Sets
- Real Financial Returns
- Linear Relationships
- Financial Econometrics and Machine Learning
- Machine Learning
- Data
- Success
- Capacity
- Evaluation
- Bias & Variance
- Cross-Validation
- AI-First Finance
- **Efficient Markets**
- Market Prediction Based on Returns Data
- Market Prediction With More Features
- Market Prediction Intraday

+ Code + Text RAM Disk Editing

Efficient Markets

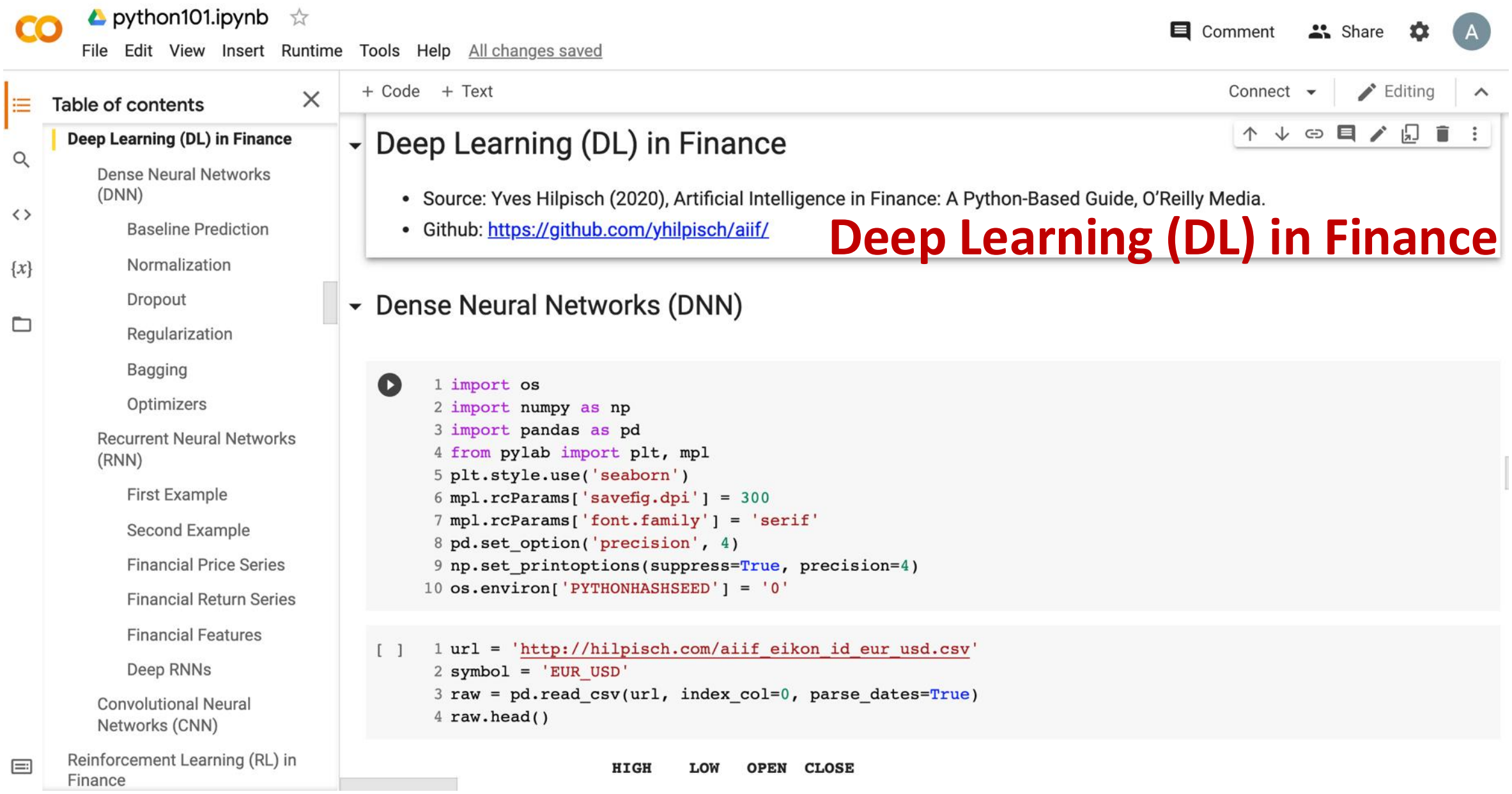
```
1 import numpy as np
2 import pandas as pd
3 from pylab import plt, mpl
4 plt.style.use('seaborn')
5 mpl.rcParams['savefig.dpi'] = 300
6 mpl.rcParams['font.family'] = 'serif'
7 pd.set_option('precision', 4)
8 np.set_printoptions(suppress=True, precision=4)
9
10 url = 'http://hilpisch.com/aiif_eikon_eod_data.csv'
11 data = pd.read_csv(url, index_col=0, parse_dates=True).dropna()
12 (data / data.iloc[0]).plot(figsize=(10, 6), cmap='coolwarm')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f29f972f210>
```

Efficient Markets

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

python101.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

Table of contents

- Deep Learning (DL) in Finance
  - Dense Neural Networks (DNN)
    - Baseline Prediction
    - Normalization
    - Dropout
    - **Regularization**
    - Bagging
    - Optimizers
  - Recurrent Neural Networks (RNN)
    - First Example
    - Second Example
    - Financial Price Series
    - Financial Return Series
    - Financial Features
    - Deep RNNs
  - Convolutional Neural Networks (CNN)
- Reinforcement Learning (RL) in Finance

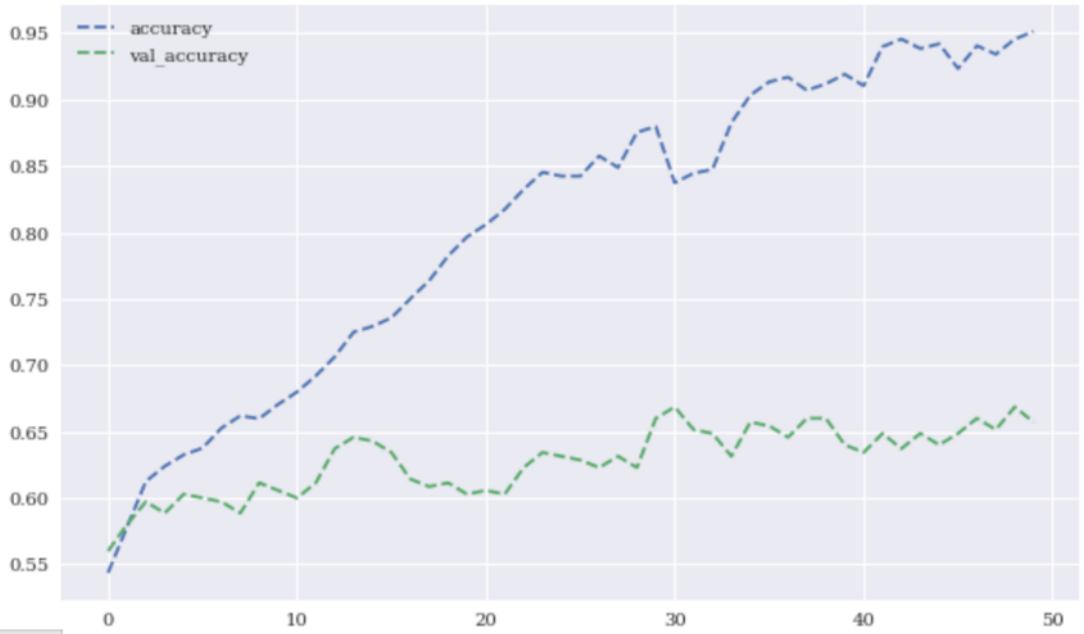
+ Code + Text Connect Editing

```
[0.35268253087997437, 0.8991981744766235]
```

```
[ ]  1 model.evaluate(test_[cols], test['d'])
```

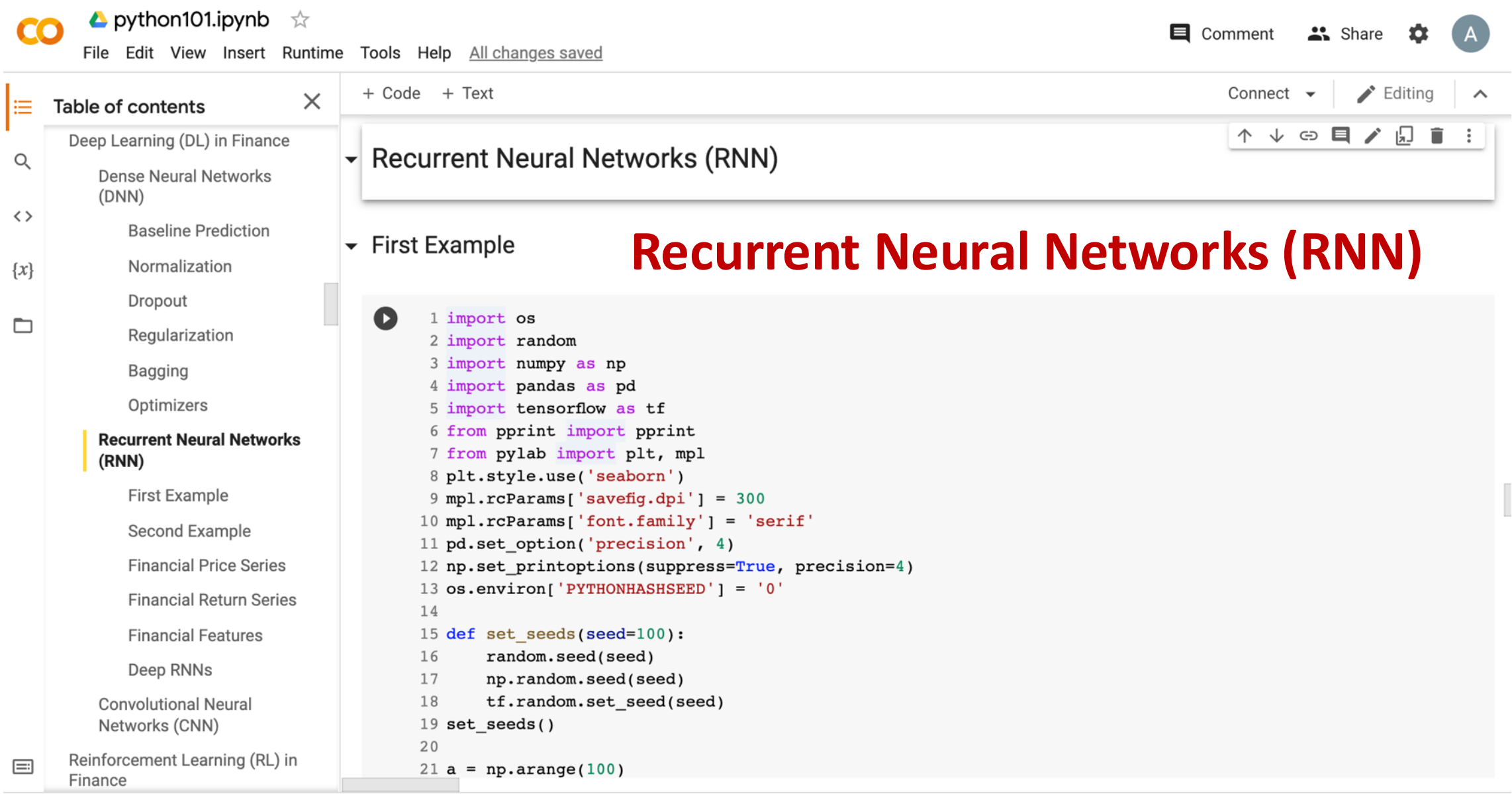
```
14/14 [==============================] - 0s 8ms/step - loss: 0.9098 - accuracy: 0.6705
[0.9098052382469177, 0.6704805493354797]
```

```
1 res = pd.DataFrame(h.history)
2 res[['accuracy', 'val_accuracy']].plot(figsize=(10, 6), style='--');
```



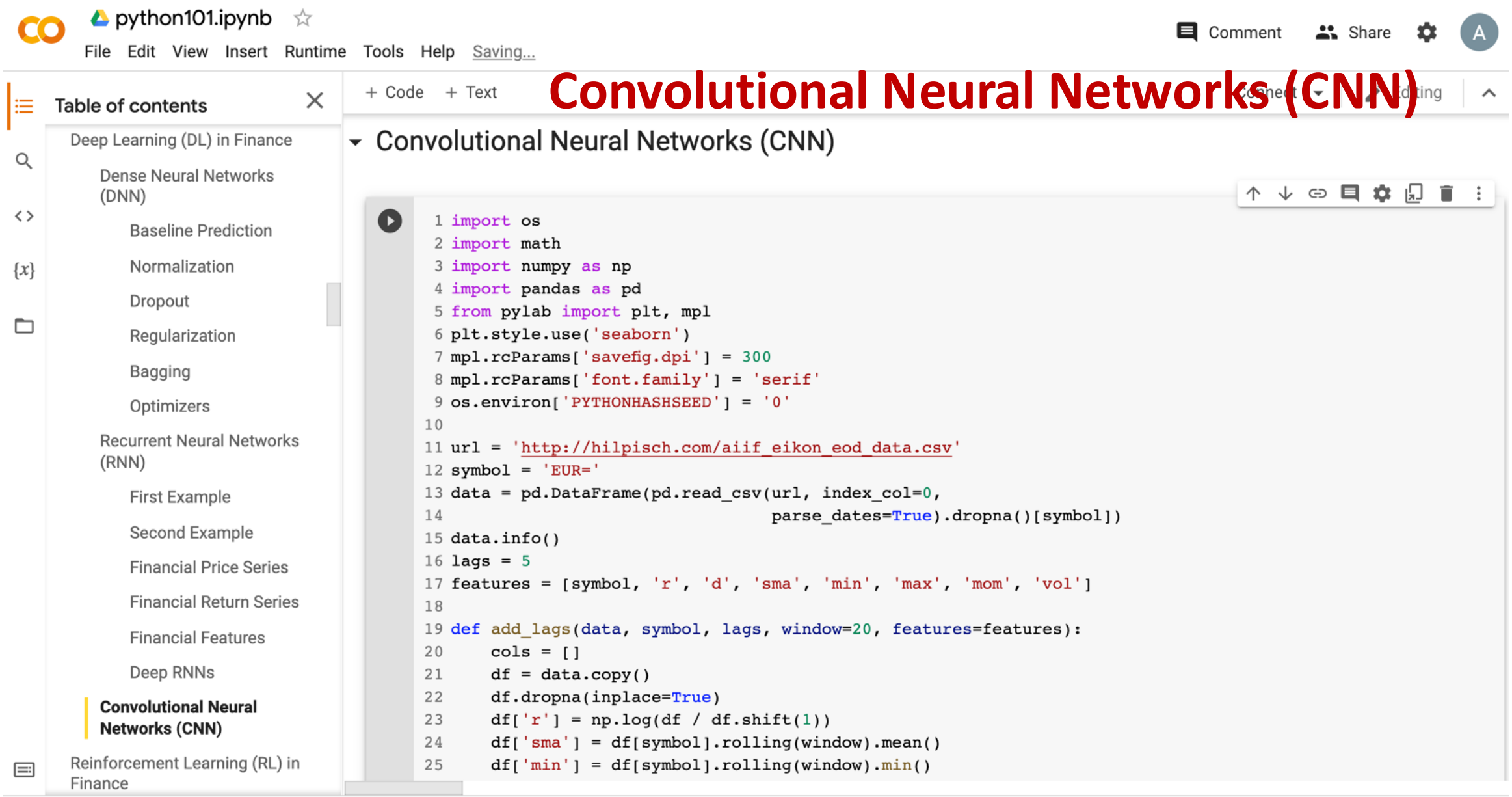
https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



https://tinyurl.com/aintpupython101

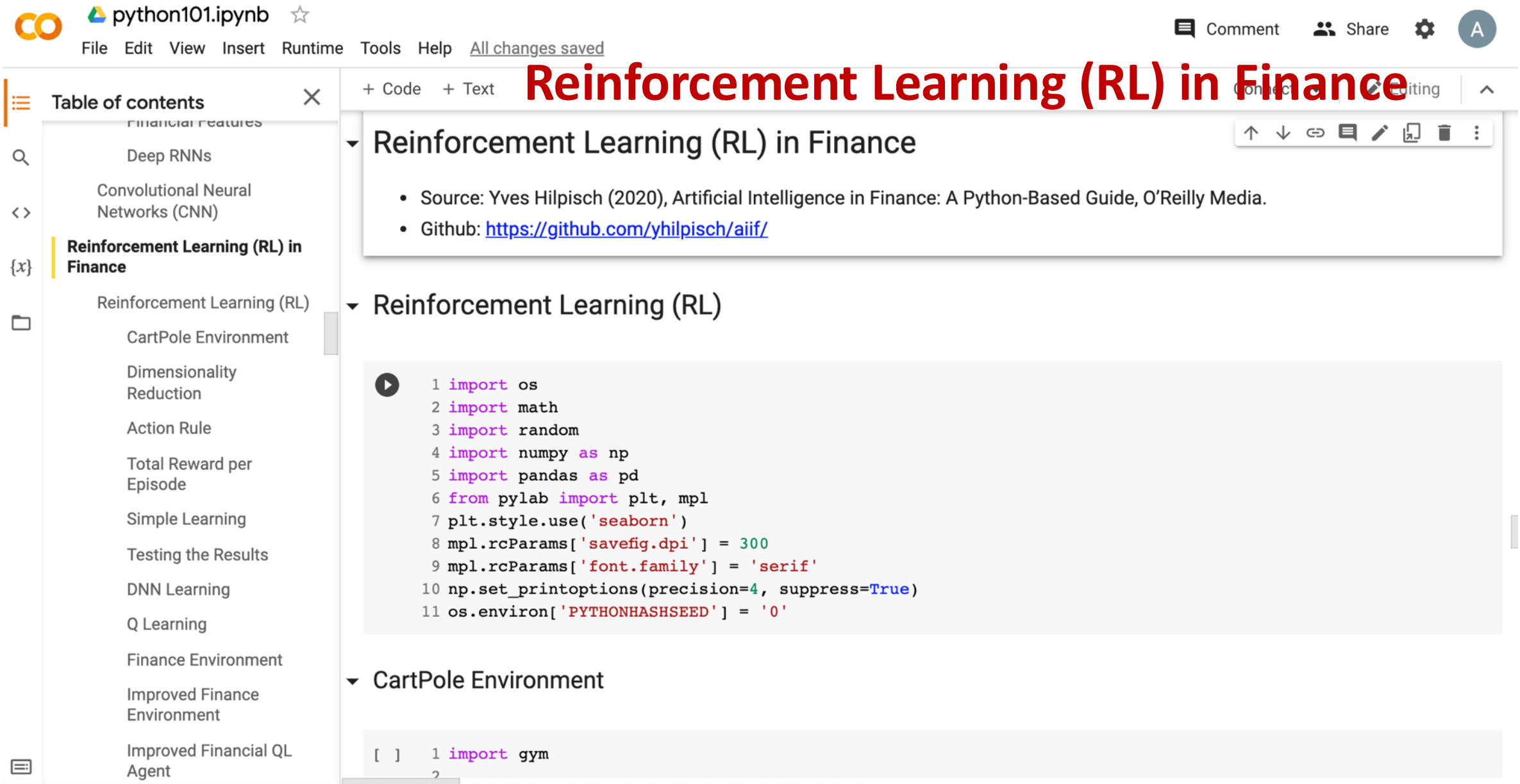
Python in Google Colab (Python101)



Convolutional Neural Networks (CNN)

https://tinyurl.com/aintpupython101

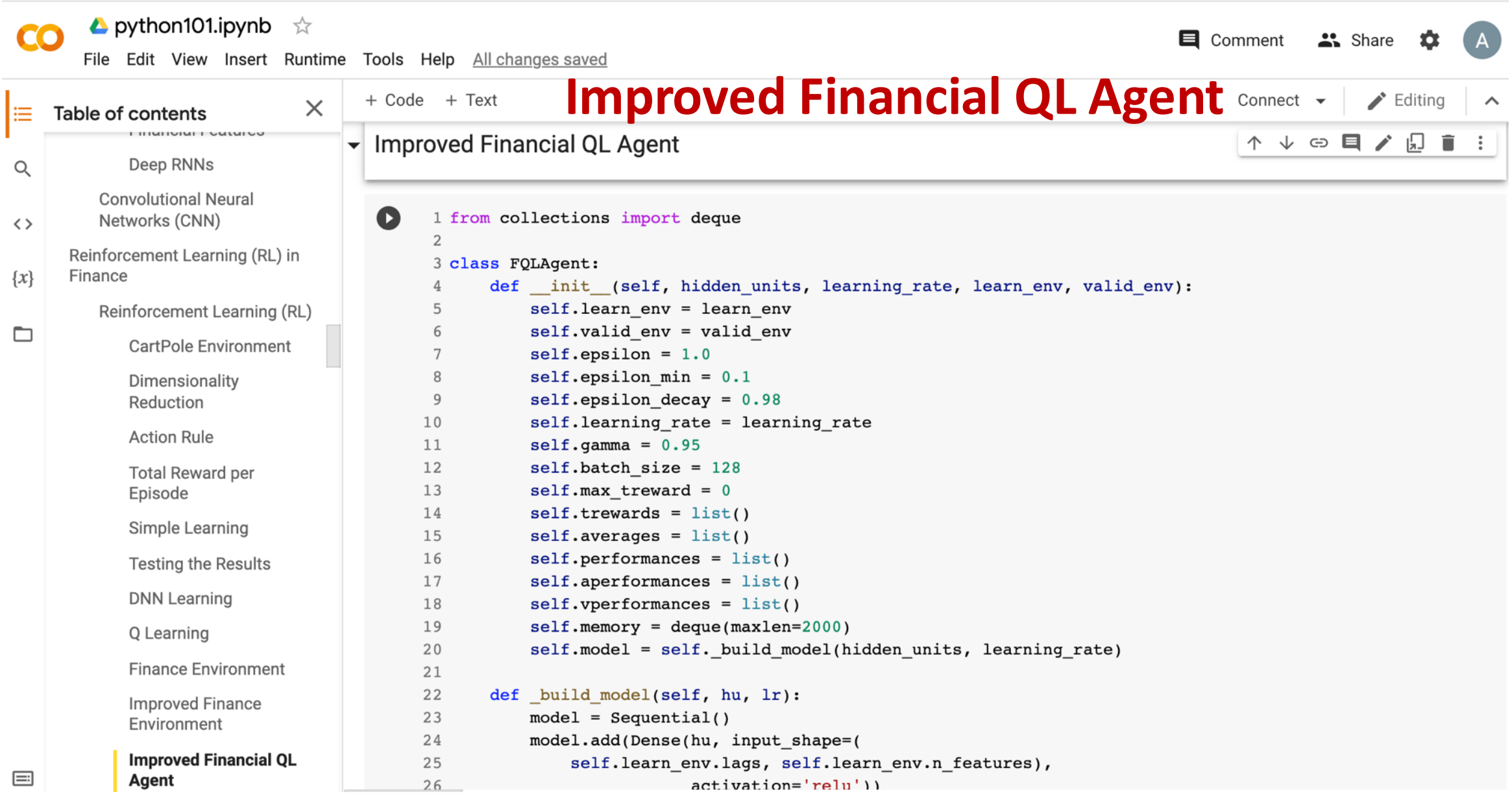
Python in Google Colab (Python101)



Reinforcement Learning (RL) in Finance

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)



Improved Financial QL Agent

https://tinyurl.com/aintpupython101

Summary

- Generative AI in Finance
- Deep Learning (DL) in Finance
  - Dense Neural Networks (DNN)
  - Recurrent Neural Networks (RNN)
  - Convolutional Neural Networks (CNN)
- Reinforcement Learning (RL) in Finance
  - Q Learning (QL)
  - Improved Finance Environment
  - Improved Financial QL Agent

Source: Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media.

References

- Siva Sai, Keya Arunakar, Vinay Chamola, Amir Hussain, Pranav Bisht, and Sanjeev Kumar (2025). "Generative AI for Finance: Applications, Case Studies and Challenges." Expert Systems 42, no. 3 (2025): e70018.
- Yuanhang Zheng, Zeshui Xu, and Anran Xiao (2023). "Deep learning in economics: a systematic and critical review." Artificial Intelligence Review (2023): 1-43.
- Ajitha Kumari Vijayappan Nair Biju, Ann Susan Thomas, and J. Thasneem (2023). "Examining the research taxonomy of artificial intelligence, deep learning & machine learning in the financial sphere—a bibliometric analysis." Quality & Quantity (2023): 1-30.
- Min-Yuh Day, Ching-Ying Yang, and Yensen Ni (2023), "Portfolio dynamic trading strategies using deep reinforcement learning." Soft Computing (2023): 1-16.
- 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.
- Yves Hilpisch (2020), Artificial Intelligence in Finance: A Python-Based Guide, O'Reilly Media, https://github.com/yhilpisch/aiif .
- 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.
- Min-Yuh Day (2025), Python 101, https://tinyurl.com/aintpupython101