人工智慧 (Artificial Intelligence)

深度學習 (Deep Learning)

1092AI08
MBA, IM, NTPU (M5010) (Spring 2021)
Wed 2, 3, 4 (9:10-12:00) (B8F40)

Min-Yuh Day
戴敏育
Associate Professor
副教授
Institute of Information Management, National Taipei University
國立臺北大學 資訊管理研究所
https://web.ntpu.edu.tw/~myday

2021-05-05

課程大綱 (Syllabus)

週次 (Week)	日期 (Date)	內容 (Subject/Topics)
1	2021/02/24	人工智慧概論 (Introduction to Artificial Intelligence)
2	2021/03/03	人工智慧和智慧代理人 (Artificial Intelligence and Intelligent Agents)
3	2021/03/10	問題解決 (Problem Solving)
4	2021/03/17	知識推理和知識表達 (Knowledge, Reasoning and Knowledge Representation)
5	2021/03/24	不確定知識和推理 (Uncertain Knowledge and Reasoning)
6	2021/03/31	人工智慧個案研究 I (Case Study on Artificial Intelligence I)

課程大綱 (Syllabus)

週次 (Week)　日期 (Date)　內容 (Subject/Topics)

7　2021/04/07　放假一天 (Day off)

8　2021/04/14　機器學習與監督式學習
(Machine Learning and Supervised Learning)

9　2021/04/21　期中報告
(Midterm Project Report)

10　2021/04/28　學習理論與綜合學習
(The Theory of Learning and Ensemble Learning)

11　2021/05/05　深度學習
(Deep Learning)

12　2021/05/12　人工智慧個案研究 II
(Case Study on Artificial Intelligence II)

課程大綱 (Syllabus)

週次 (Week) 日期 (Date) 內容 (Subject/Topics)

13 2021/05/19 強化學習 (Reinforcement Learning)

14 2021/05/26 深度學習自然語言處理 (Deep Learning for Natural Language Processing)

15 2021/06/02 機器人技術 (Robotics)

16 2021/06/09 人工智慧哲學與倫理，人工智慧的未來 (Philosophy and Ethics of AI, The Future of AI)

17 2021/06/16 期末報告 I (Final Project Report I)

18 2021/06/23 期末報告 II (Final Project Report II)

Deep Learning

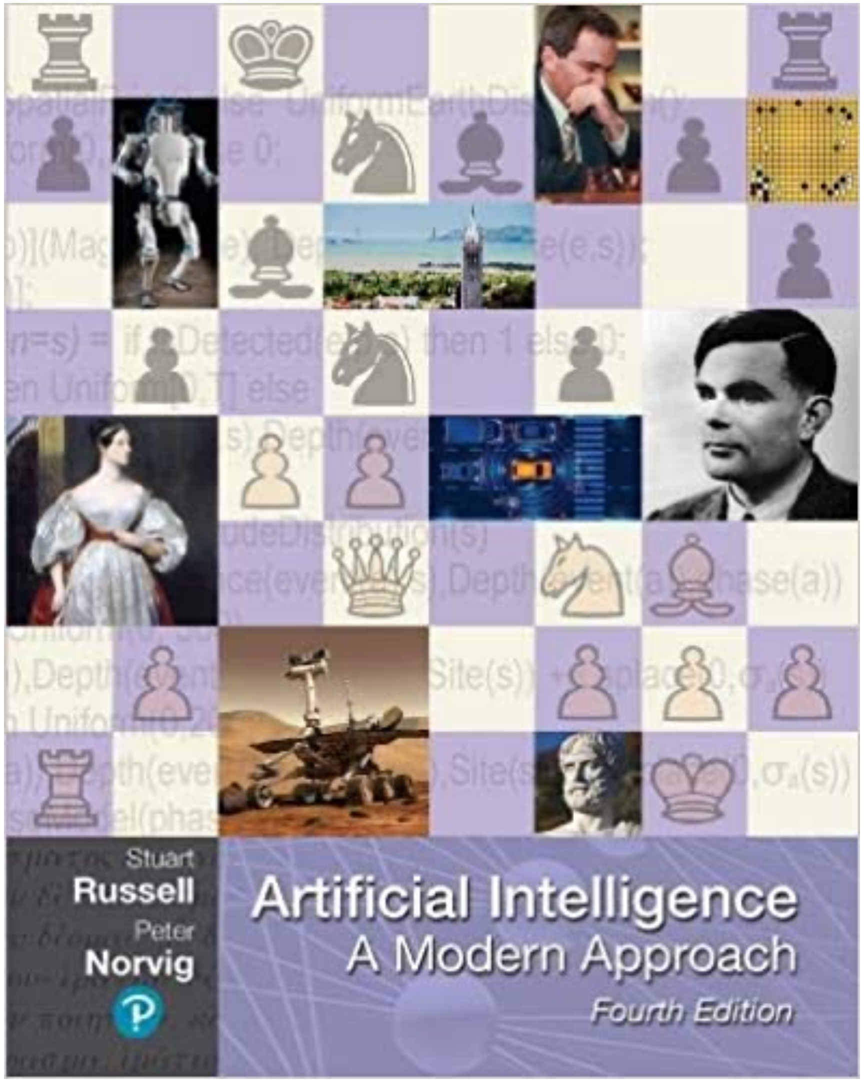
Outline

- **Deep Learning**
- **Neural Networks (NN)**
- **Convolutional Neural Networks (CNN)**
- **Recurrent Neural Networks (RNN)**

Stuart Russell and Peter Norvig (2020),

Artificial Intelligence: A Modern Approach,

4th Edition, Pearson



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

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

Artificial Intelligence: A Modern Approach

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

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

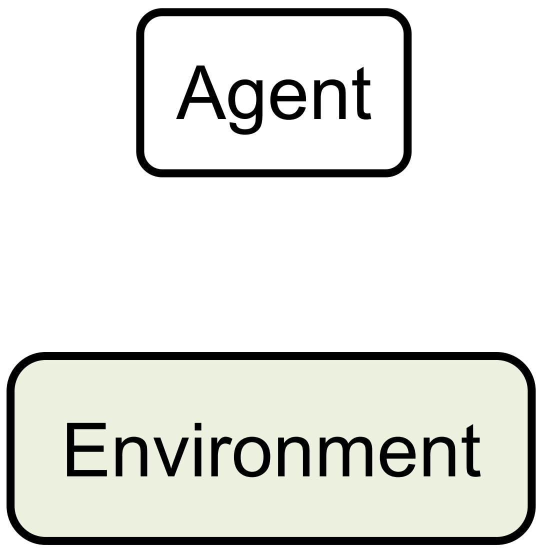
Artificial Intelligence:
Machine Learning

Artificial Intelligence: 5. Machine Learning

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

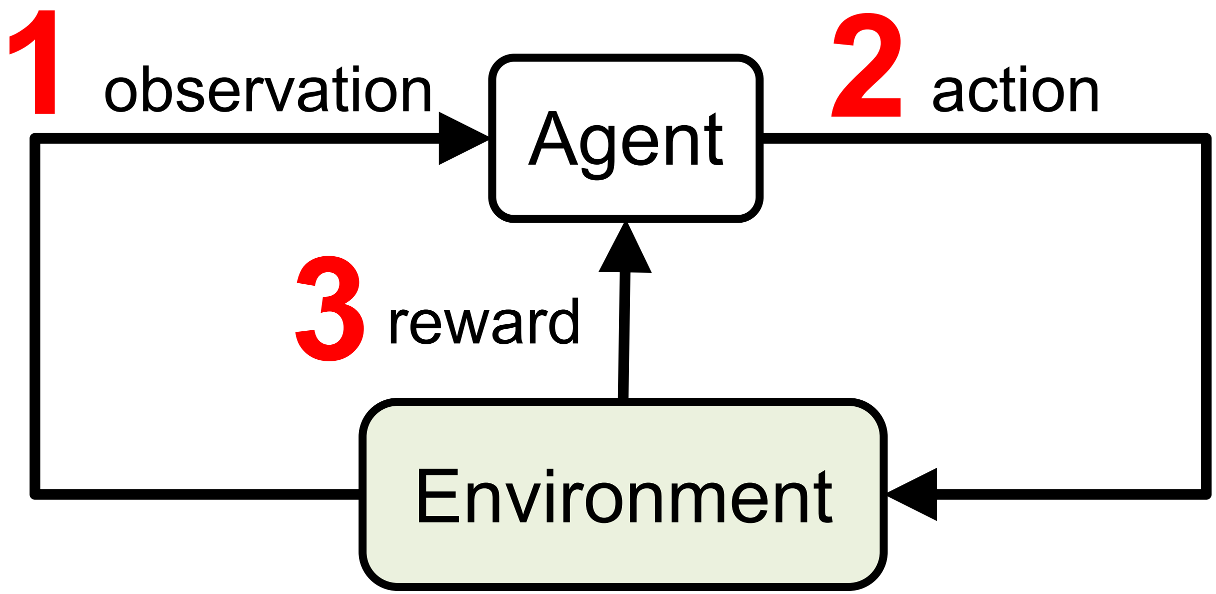
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Reinforcement Learning (DL)



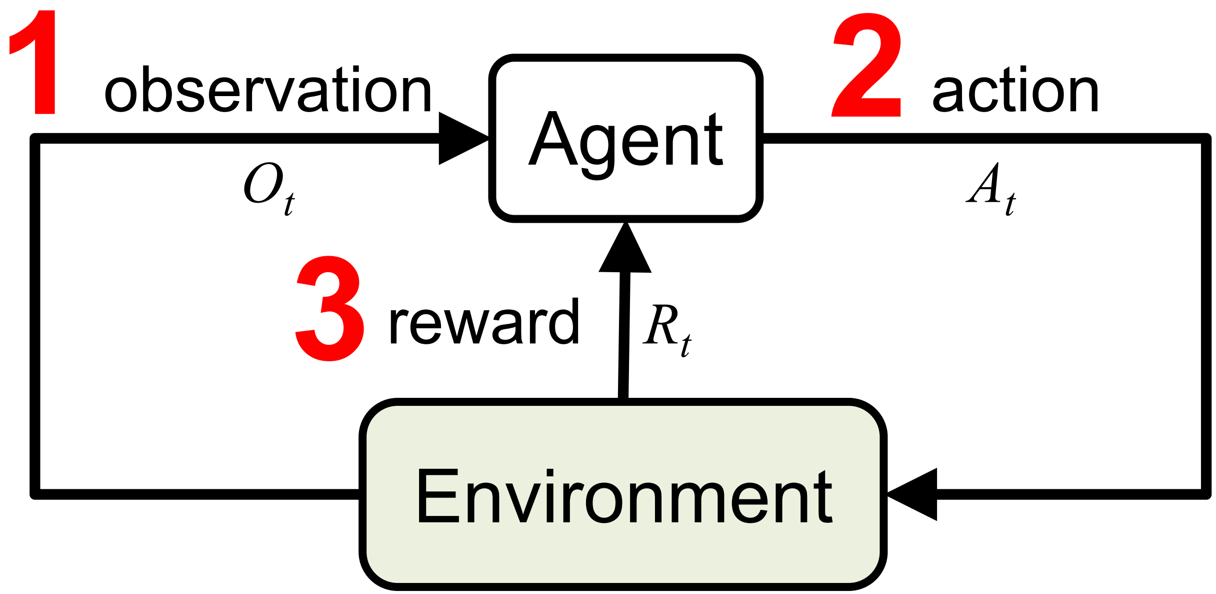
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford

Reinforcement Learning (DL)



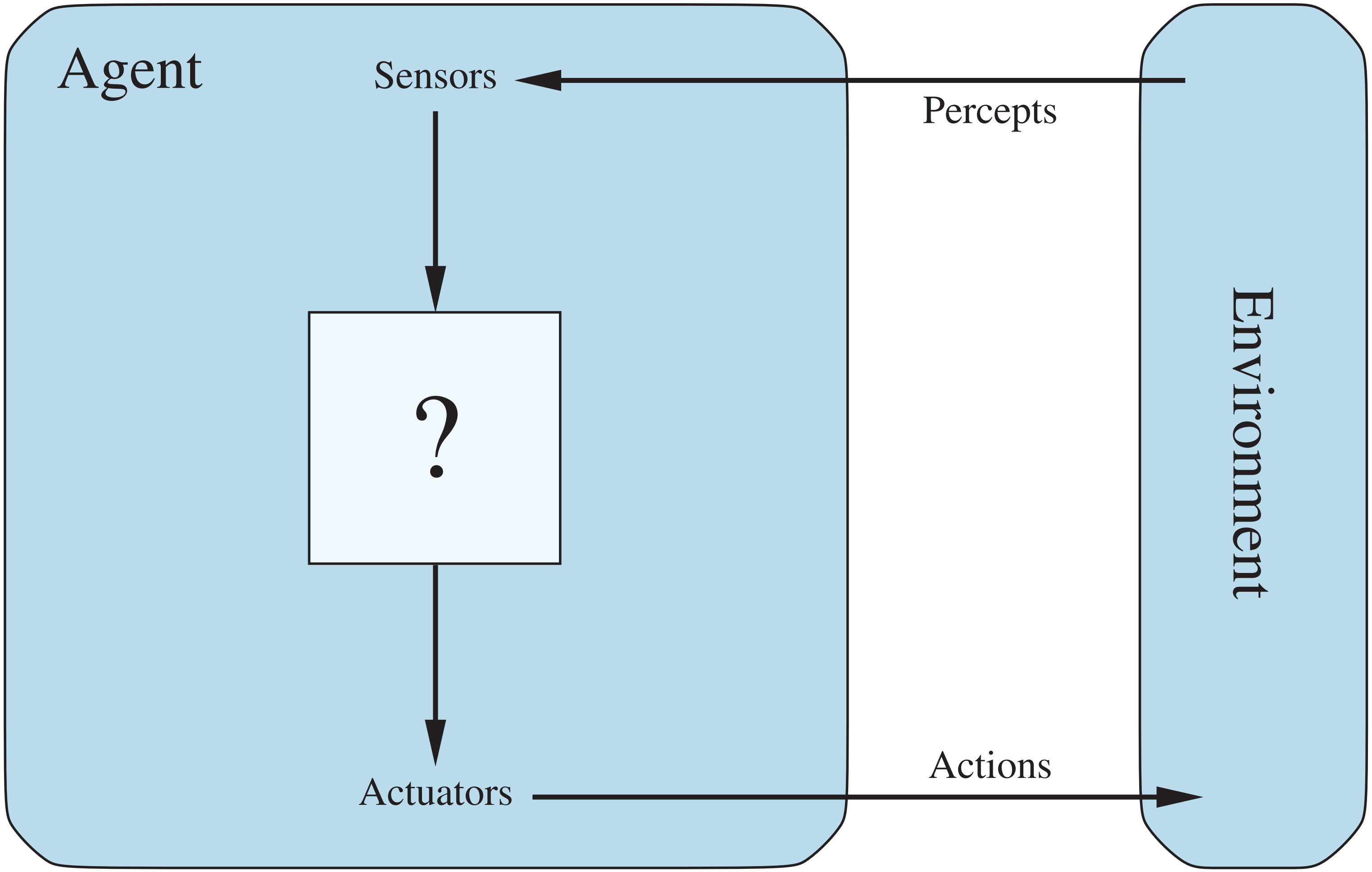
Source: Richard S. Sutton & Andrew G. Barto (2018), Reinforcement Learning: An Introduction, 2nd Edition, A Bradford

Reinforcement Learning (DL)



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

Agents interact with environments through sensors and actuators



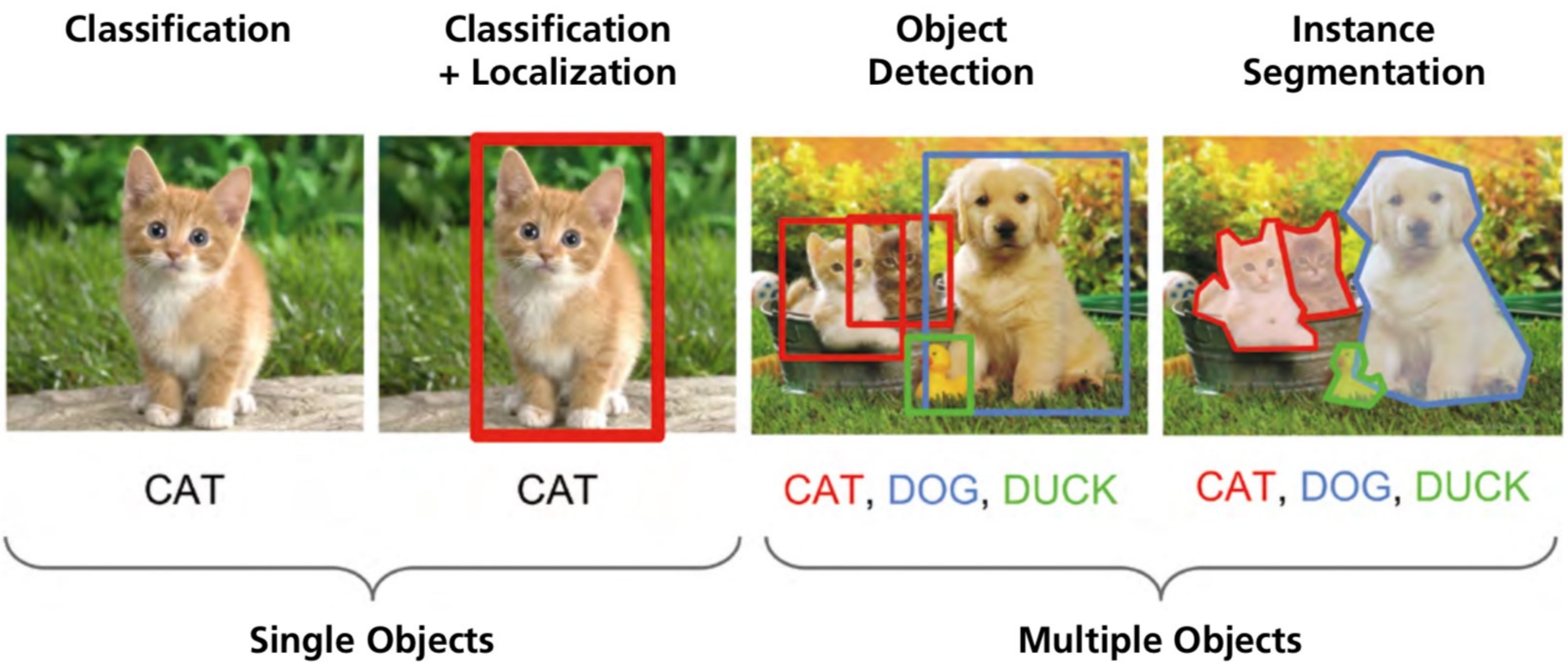
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

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

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

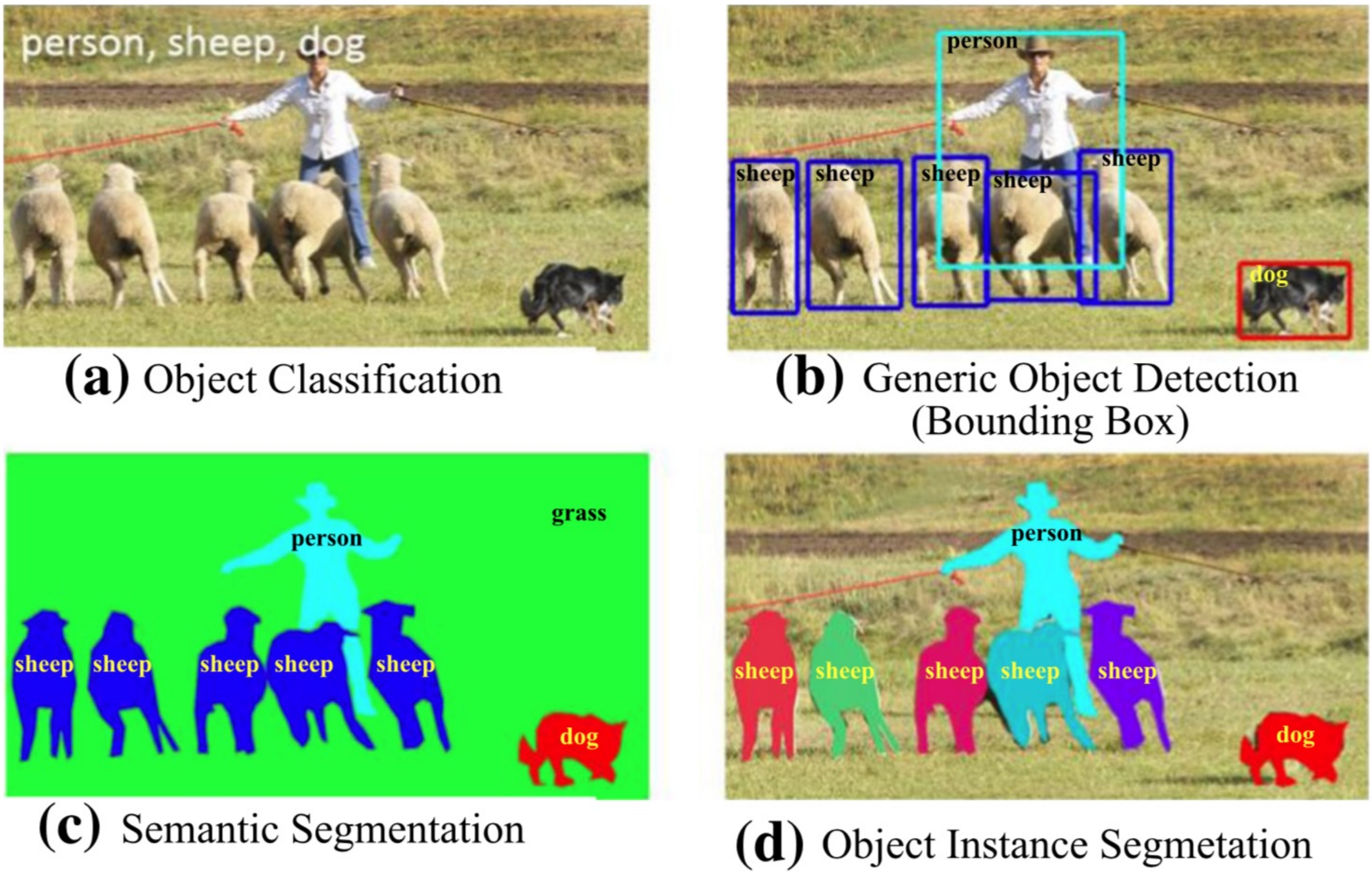
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Computer Vision: Image Classification, Object Detection, Object Instance Segmentation



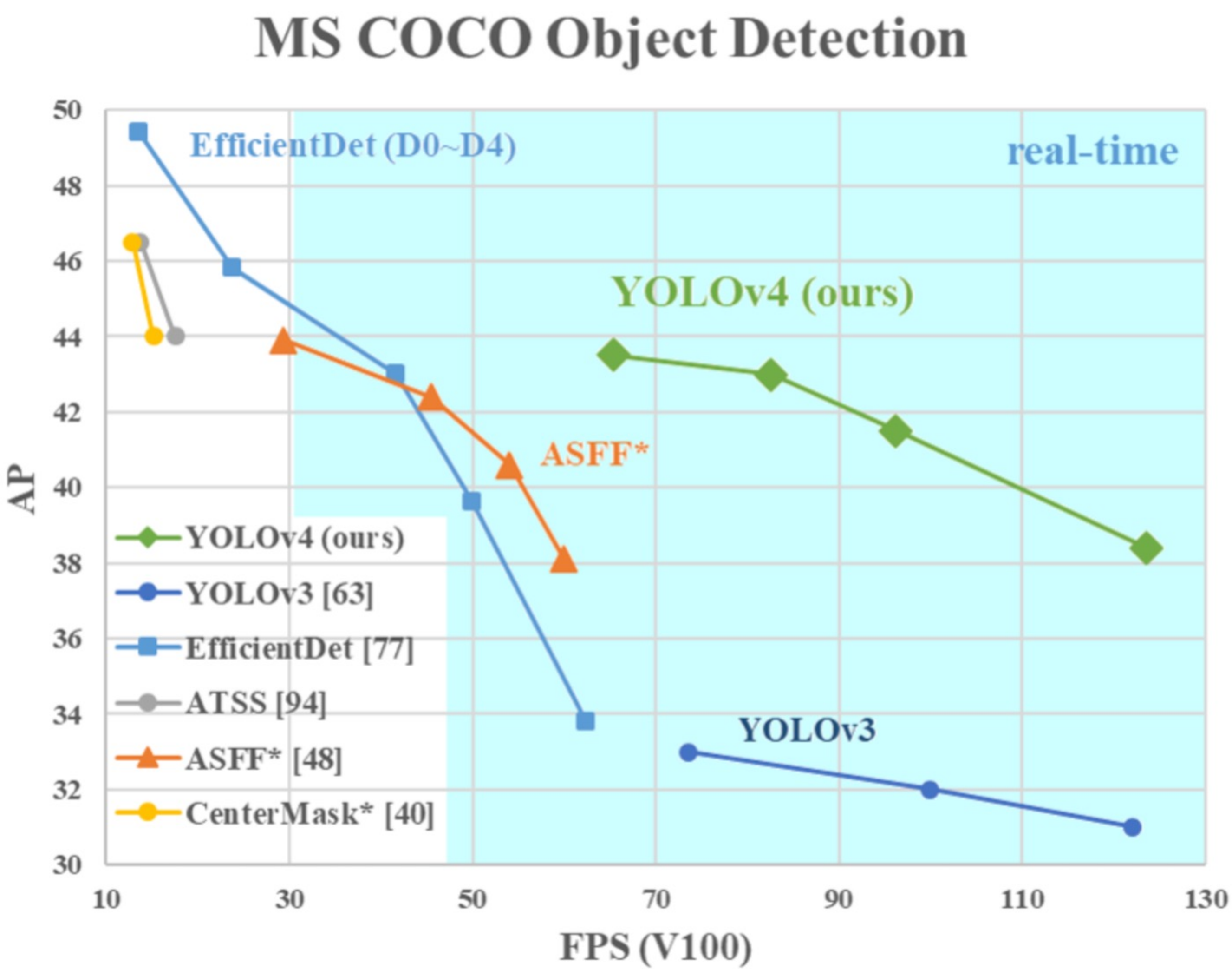
Source: DHL (2018), Artificial Intelligence in Logistics,
http://www.globalhha.com/doclib/data/upload/doc_con/5e50c53c5bf67.pdf/

Computer Vision: Object Detection



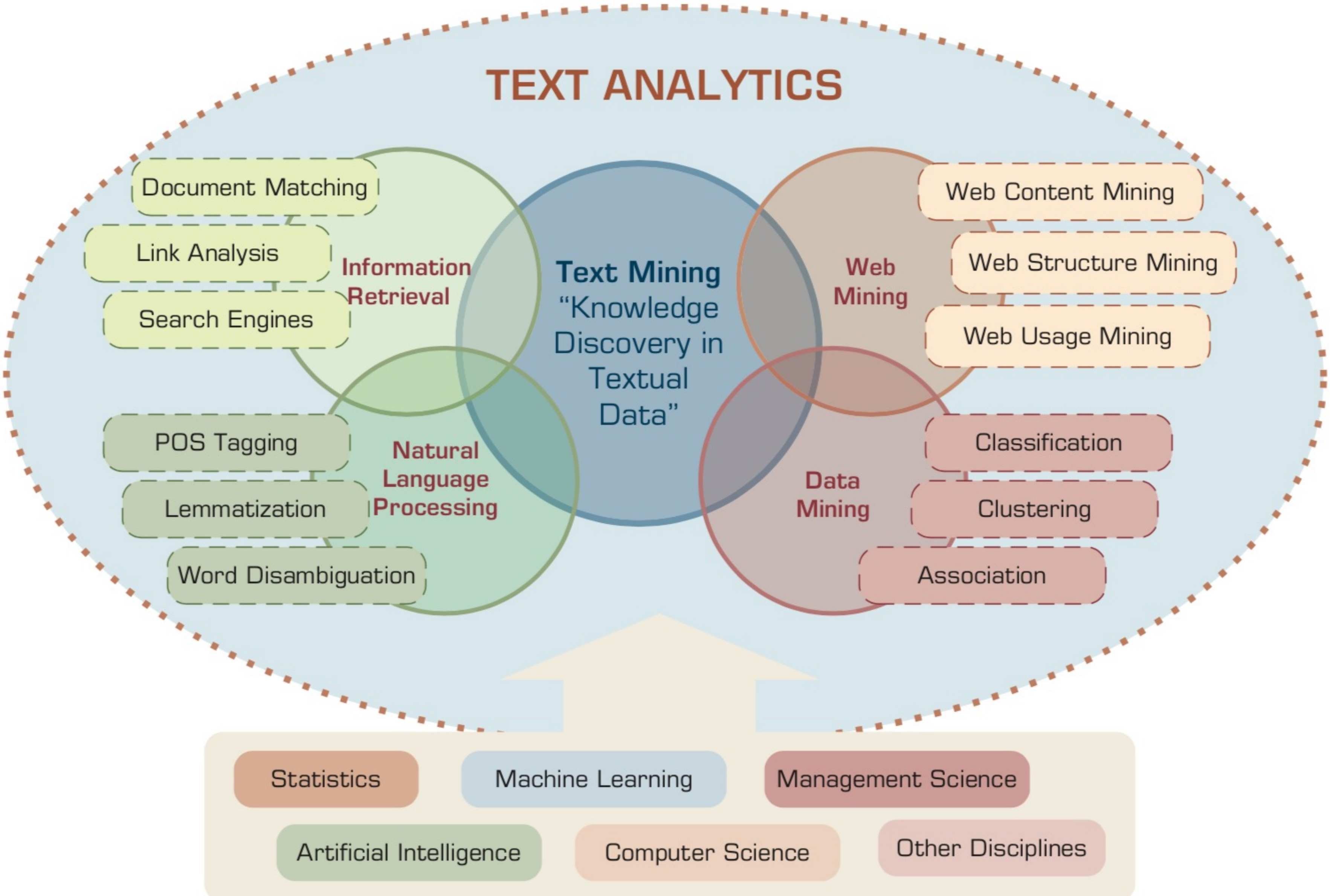
Source: Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. "Deep learning for generic object detection: A survey." International journal of computer vision 128, no. 2 (2020): 261-318.

YOLOv4:
Optimal Speed and Accuracy of Object Detection



Source: Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection." arXiv preprint arXiv:2004.10934 (2020).

Text Analytics and Text Mining



Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

Deep learning for financial applications: A survey

Applied Soft Computing (2020)

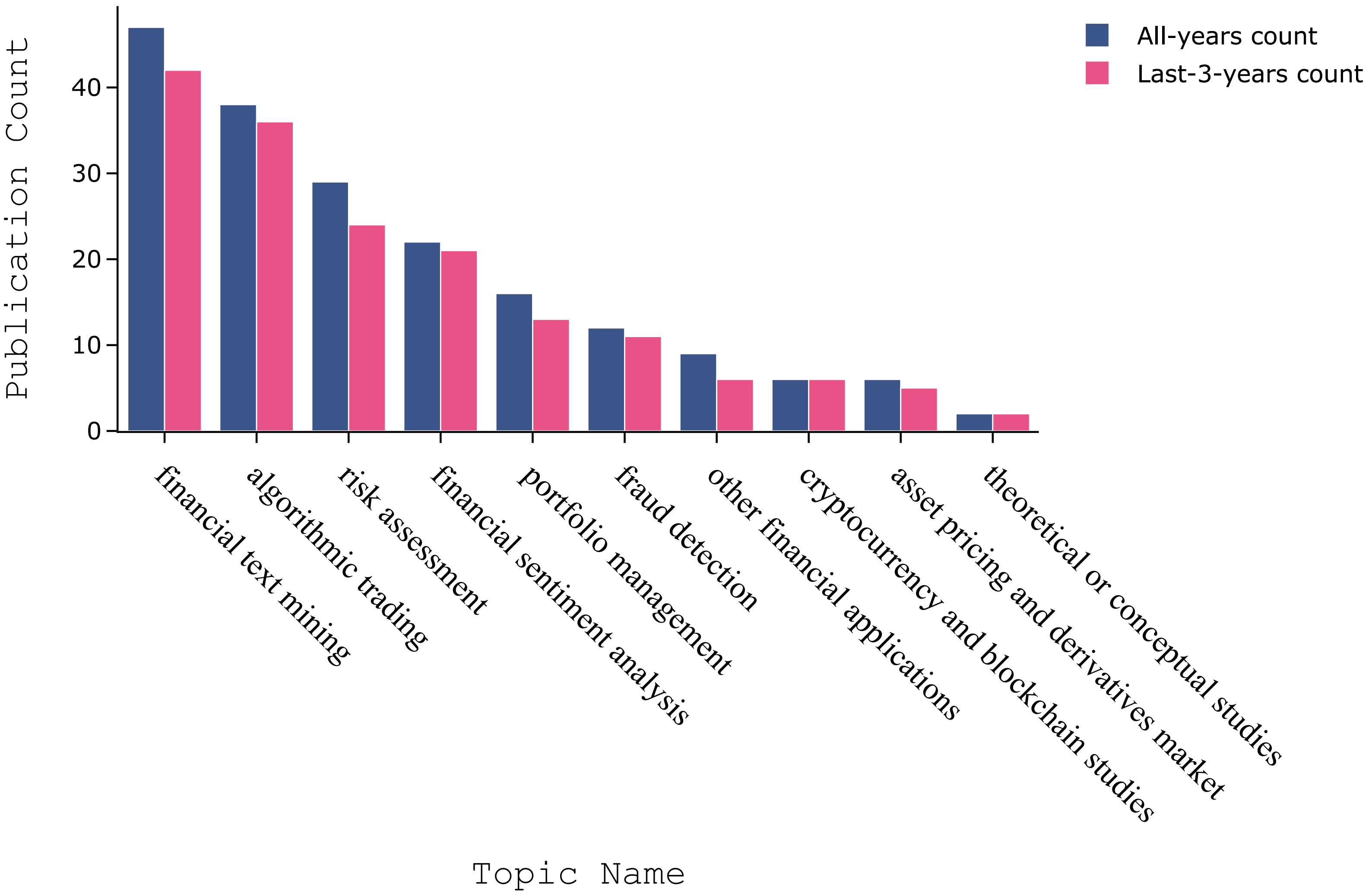
Source:
Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020).
"Deep learning for financial applications: A survey."
Applied Soft Computing (2020): 106384.

Financial time series forecasting with deep learning: A systematic literature review: 2005–2019

Applied Soft Computing (2020)

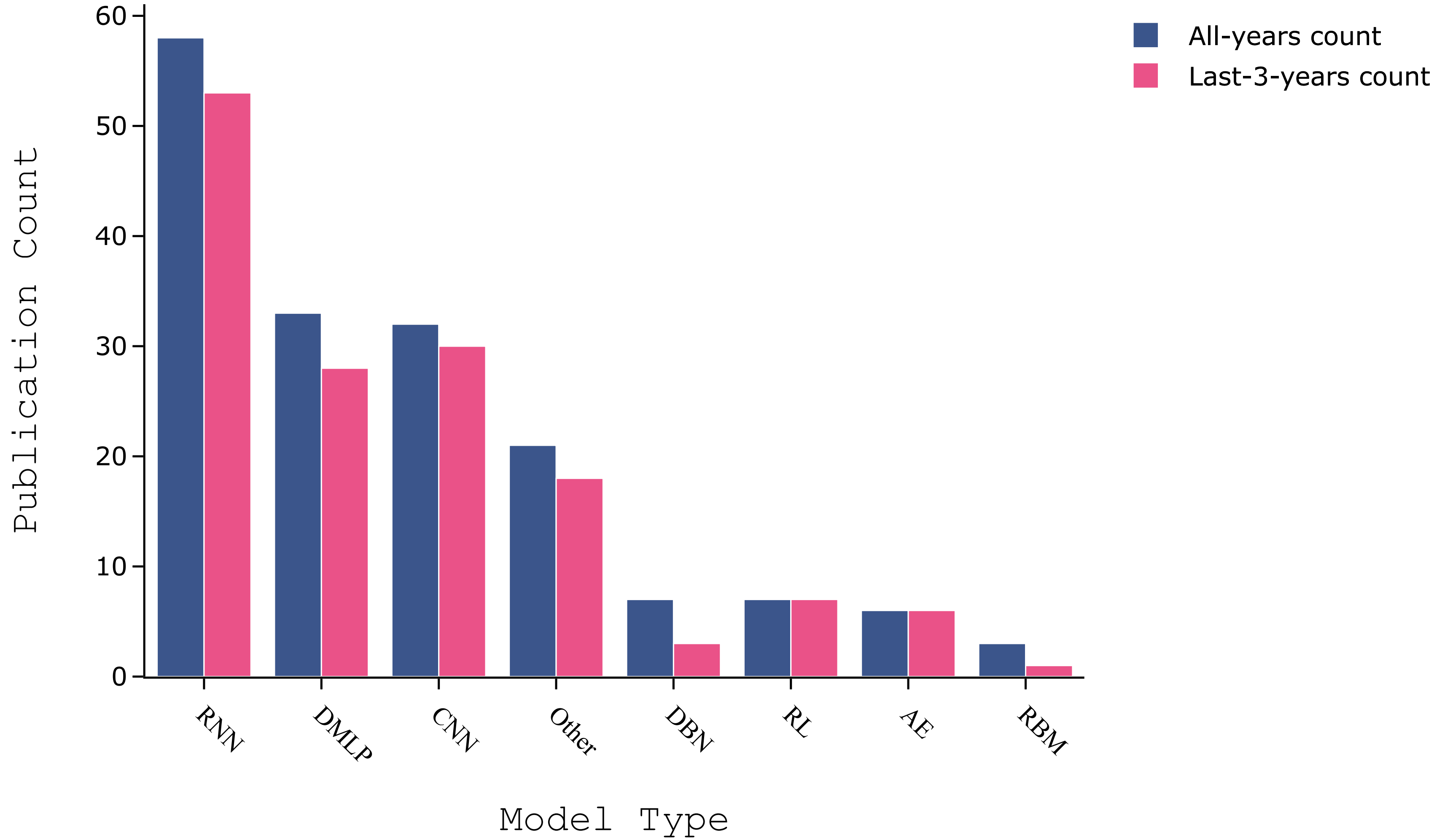
Source:
Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu (2020), "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019." Applied Soft Computing 90 (2020): 106181.

Deep learning for financial applications: Topics



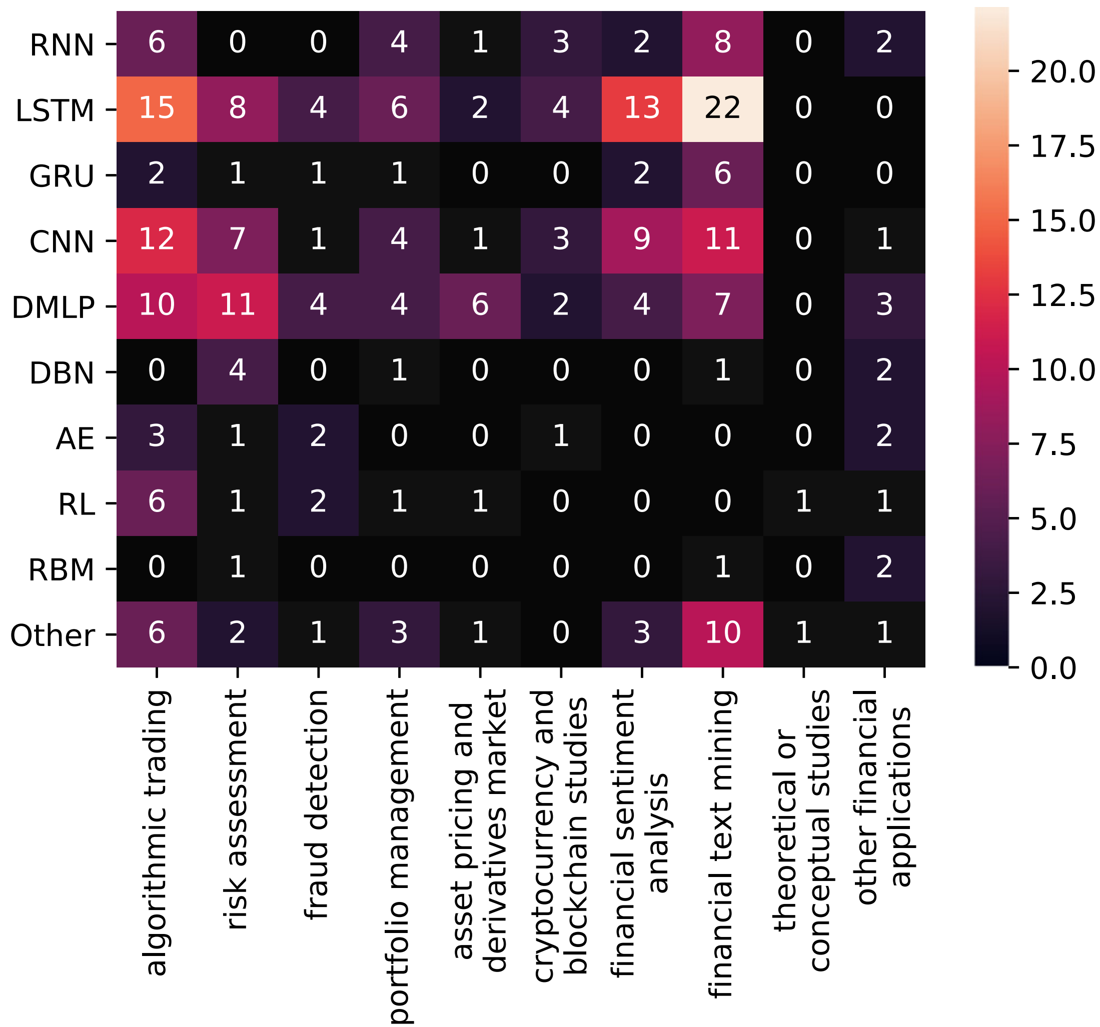
Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications: Deep Learning Models



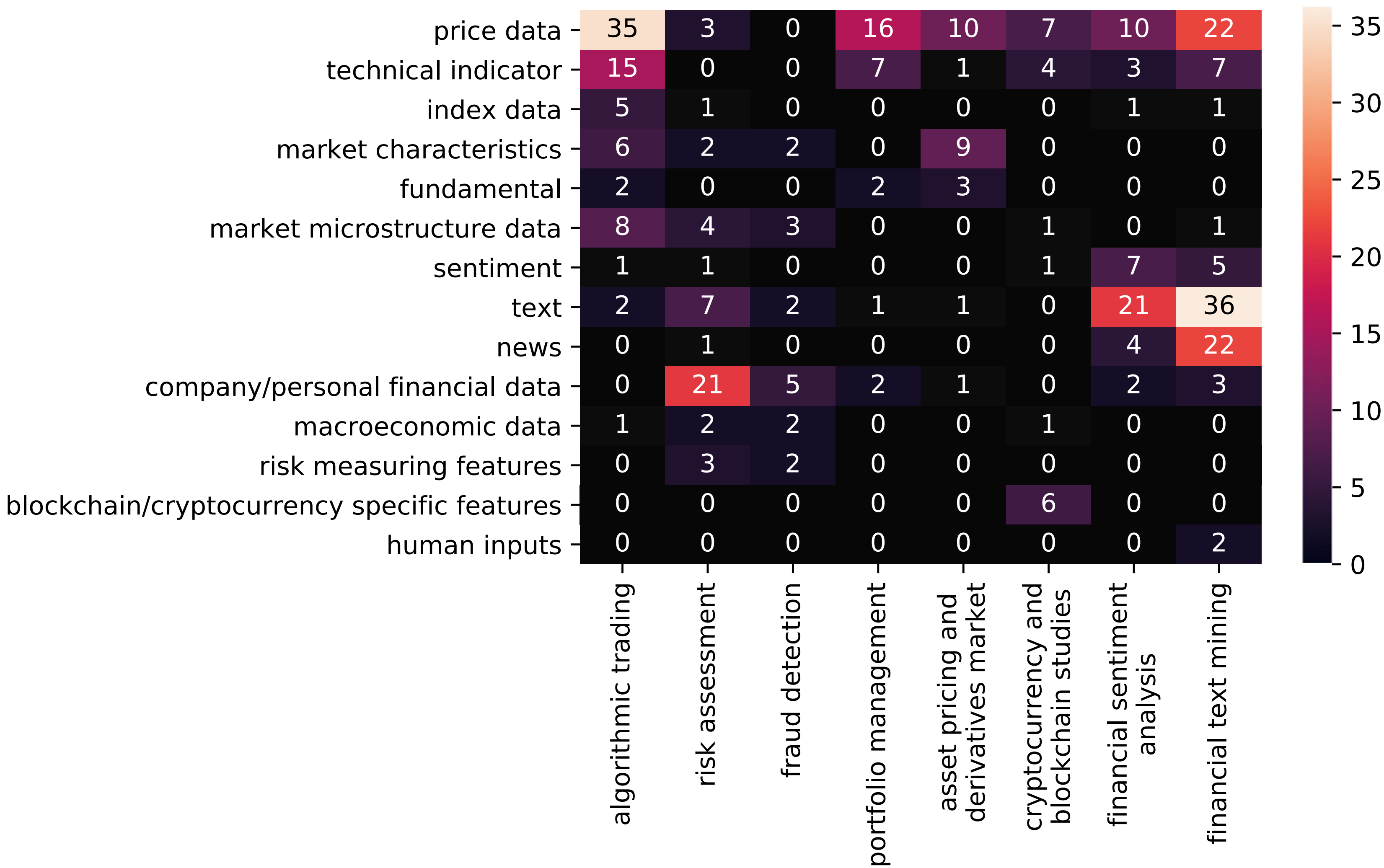
Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications: Topic-Model Heatmap

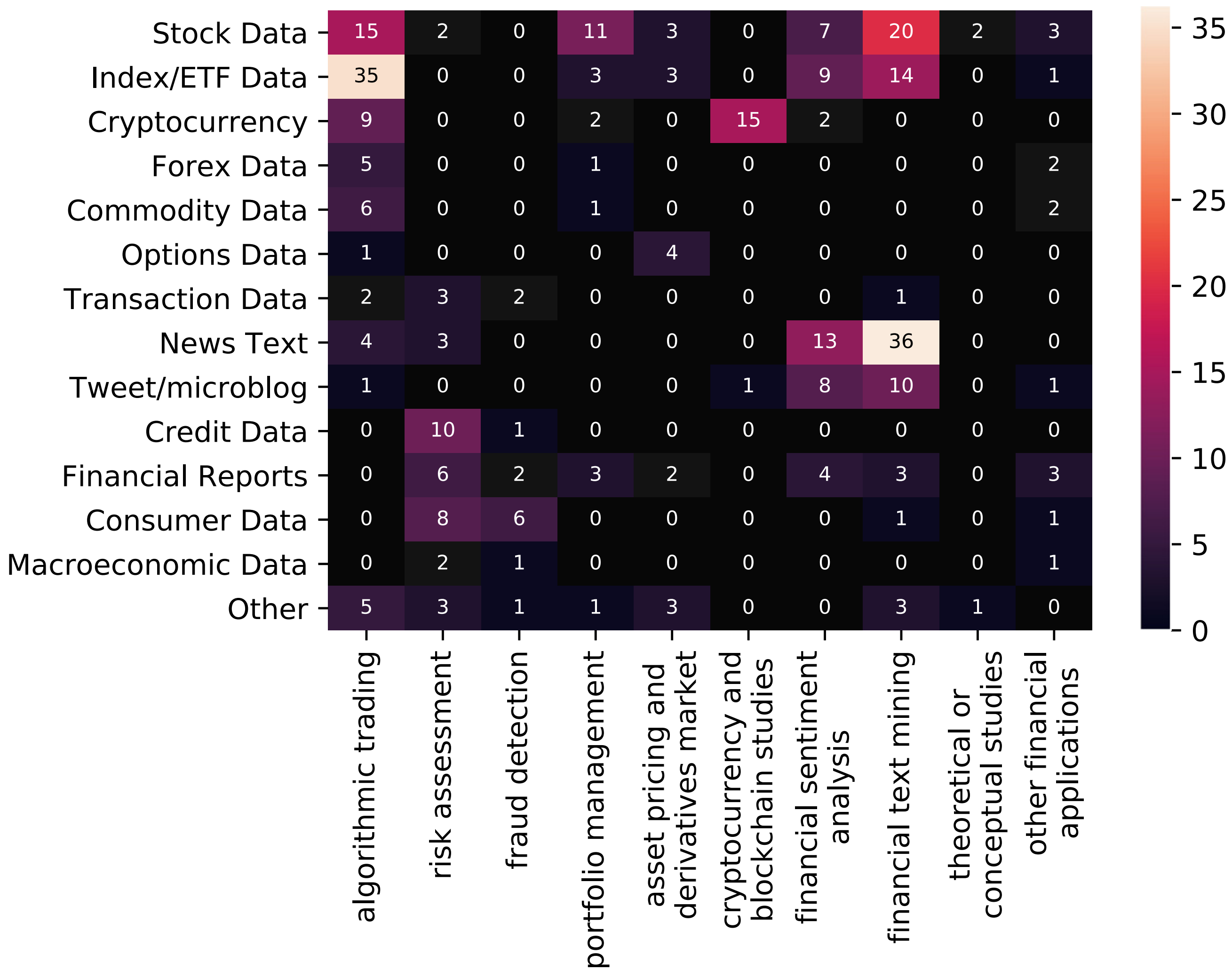


Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

Deep learning for financial applications: Topic-Feature Heatmap



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.

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

Art.	Data set	Period	Feature set	Method	Performance criteria	Env.
[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 theoretical or conceptual studies

Art.	SubTopic	IsTimeSeries?	Data set	Period	Feature set	Method
[197]	Analysis of AE, SVD	Yes	Selected stocks from the IBB index and stock of Amgen Inc.	2012–2014	Price data	AE, SVD
[198]	Fraud Detection in Banking	No	Risk Management / Fraud Detection	–	–	DRL

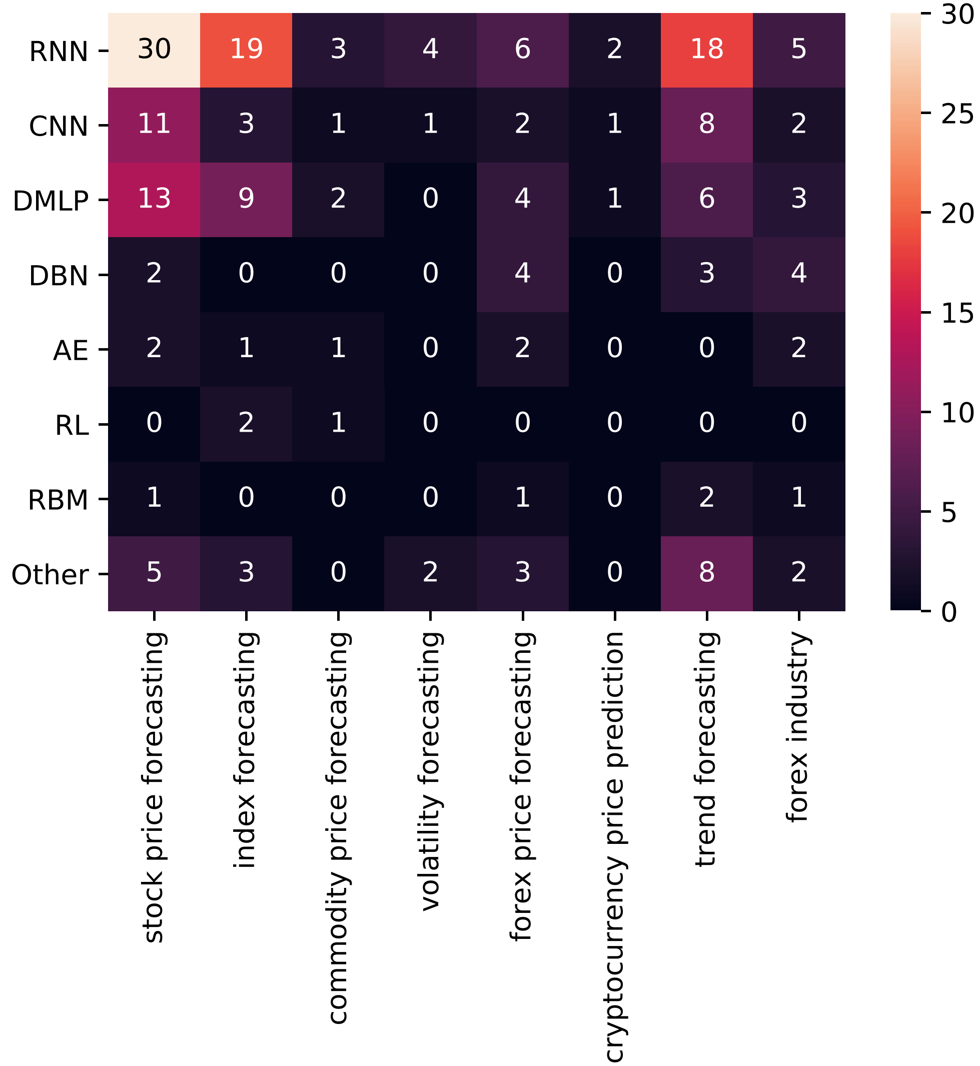
Source: Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.

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.

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.

Stock price forecasting using only raw time series data

Art.	Data set	Period	Feature set	Lag	Horizon	Method	Performance criteria	Env.
[80]	38 stocks in KOSPI	2010–2014	Lagged stock returns	50 min	5 min	DNN	NMSE, RMSE, MAE, MI	–
[81]	China stock market, 3049 Stocks	1990–2015	OCHLV	30 d	3 d	LSTM	Accuracy	Theano, Keras
[82]	Daily returns of 'BRD' stock in Romanian Market	2001–2016	OCHLV	–	1 d	LSTM	RMSE, MAE	Python, Theano
[83]	297 listed companies of CSE	2012–2013	OCHLV	2 d	1 d	LSTM, SRNN, GRU	MAD, MAPE	Keras
[84]	5 stock in NSE	1997–2016	OCHLV, Price data, turnover and number of trades.	200 d	1..10 d	LSTM, RNN, CNN, MLP	MAPE	–
[85]	Stocks of Infosys, TCS and CIPLA from NSE	2014	Price data	–	–	RNN, LSTM and CNN	Accuracy	–
[86]	10 stocks in S&P500	1997–2016	OCHLV, Price data	36 m	1 m	RNN, LSTM, GRU	Accuracy, Monthly return	Keras, Tensorflow
[87]	Stocks data from S&P500	2011–2016	OCHLV	1 d	1 d	DBN	MSE, norm-RMSE, MAE	–
[88]	High-frequency transaction data of the CSI300 futures	2017	Price data	–	1 min	DNN, ELM, RBF	RMSE, MAPE, Accuracy	Matlab
[89]	Stocks in the S&P500	1990–2015	Price data	240 d	1 d	DNN, GBT, RF	Mean return, MDD, Calmar ratio	H2O
[90]	ACI Worldwide, Staples, and Seagate in NASDAQ	2006–2010	Daily closing prices	17 d	1 d	RNN, ANN	RMSE	–
[91]	Chinese Stocks	2007–2017	OCHLV	30 d	1..5 d	CNN + LSTM	Annualized Return, Mxm Retracement	Python
[92]	20 stocks in S&P500	2010–2015	Price data	–	–	AE + LSTM	Weekly Returns	–
[93]	S&P500	1985–2006	Monthly and daily log-returns	*	1 d	DBN+MLP	Validation, Test Error	Theano, Python, Matlab
[94]	12 stocks from SSE Composite Index	2000–2017	OCHLV	60 d	1..7 d	DWNN	MSE	Tensorflow
[95]	50 stocks from NYSE	2007–2016	Price data	–	1d, 3 d, 5 d	SFM	MSE	–

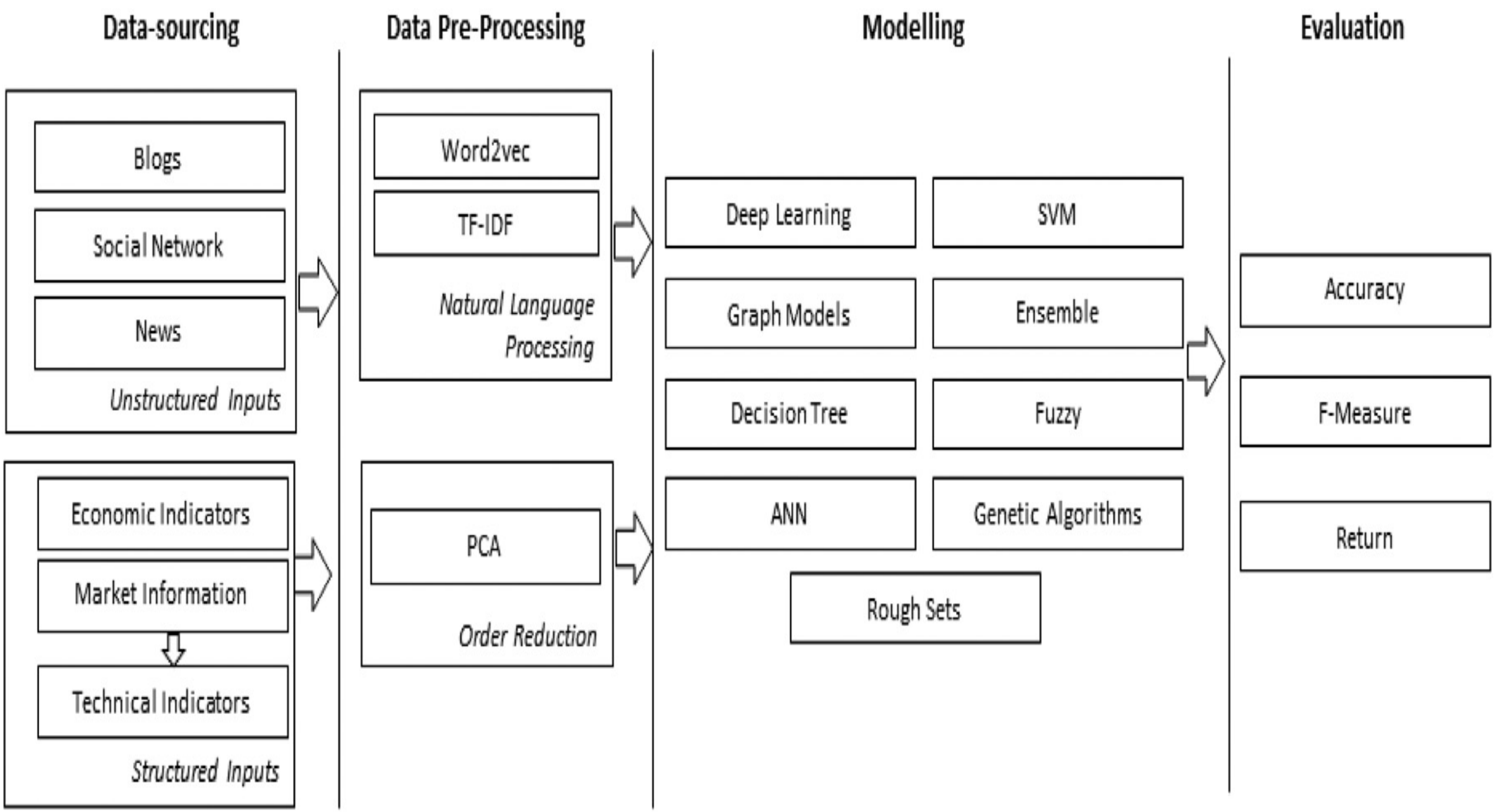
Source: Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu (2020), "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019." Applied Soft Computing 90 (2020): 106181.

Stock price forecasting using various data

Art.	Data set	Period	Feature set	Lag	Horizon	Method	Performance criteria	Env.
[96]	Japan Index constituents from WorldScope	1990–2016	25 Fundamental Features	10 d	1 d	DNN	Correlation, Accuracy, MSE	Tensorflow
[97]	Return of S&P500	1926–2016	Fundamental Features:	–	1 s	DNN	MSPE	Tensorflow
[98]	U.S. low-level disaggregated macroeconomic time series	1959–2008	GDP, Unemployment rate, Inventories, etc.	–	–	DNN	R^2	–
[99]	CDAX stock market data	2010–2013	Financial news, stock market data	20 d	1 d	LSTM	MSE, RMSE, MAE, Accuracy, AUC	TensorFlow, Theano, Python, Scikit-Learn
[100]	Stock of Tsugami Corporation	2013	Price data	–	–	LSTM	RMSE	Keras, Tensorflow
[101]	Stocks in China's A-share	2006–2007	11 technical indicators	–	1 d	LSTM	AR, IR, IC	–
[102]	SCI prices	2008–2015	OCHL of change rate, price	7 d	–	EmotionalAnalysis + LSTM	MSE	–
[103]	10 stocks in Nikkei 225 and news	2001–2008	Textual information and Stock prices	10 d	–	Paragraph Vector + LSTM	Profit	–
[104]	TKC stock in NYSE and QQQQ ETF	1999–2006	Technical indicators, Price	50 d	1 d	RNN (Jordan–Elman)	Profit, MSE	Java
[105]	10 Stocks in NYSE	–	Price data, Technical indicators	20 min	1 min	LSTM, MLP	RMSE	–
[106]	42 stocks in China's SSE	2016	OCHLV, Technical Indicators	242 min	1 min	GAN (LSTM, CNN)	RMSRE, DPA, GAN-F, GAN-D	–
[107]	Google's daily stock data	2004–2015	OCHLV, Technical indicators	20 d	1 d	$(2D)^2$ PCA + DNN	SMAPE, PCD, MAPE, RMSE, HR, TR, R^2	R, Matlab
[108]	GarantiBank in BIST, Turkey	2016	OCHLV, Volatility, etc.	–	–	PLR, Graves LSTM	MSE, RMSE, MAE, RSE, R^2	Spark
[109]	Stocks in NYSE, AMEX, NASDAQ, TAQ intraday trade	1993–2017	Price, 15 firm characteristics	80 d	1 d	LSTM+MLP	Monthly return, SR	Python,Keras, Tensorflow in AWS
[110]	Private brokerage company's real data of risky transactions	–	250 features: order details, etc.	–	–	CNN, LSTM	F1-Score	Keras, Tensorflow
[111]	Fundamental and Technical Data, Economic Data	–	Fundamental , technical and market information	–	–	CNN	–	–
[112]	The LOB of 5 stocks of Finnish Stock Market	2010	FI-2010 dataset: bid/ask and volume	–	*	WMTR, MDA	Accuracy, Precision, Recall, F1-Score	–
[113]	Returns in NYSE, AMEX, NASDAQ	1975–2017	57 firm characteristics	*	–	Fama–French n-factor model DL	R^2, RMSE	Tensorflow

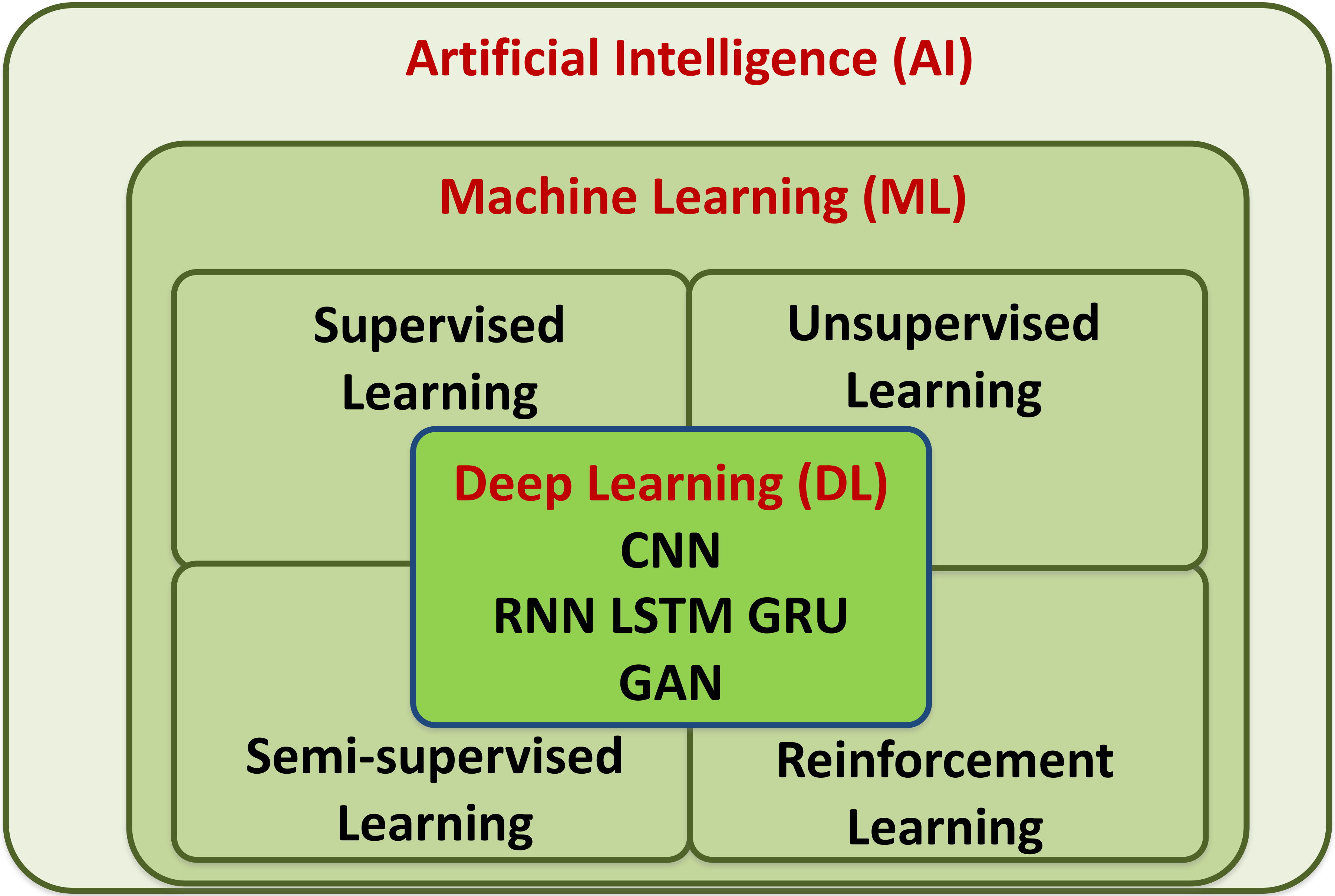
Source: Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu (2020), "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019." Applied Soft Computing 90 (2020): 106181.

Stock Market Movement Forecast:
Phases of the stock market modeling



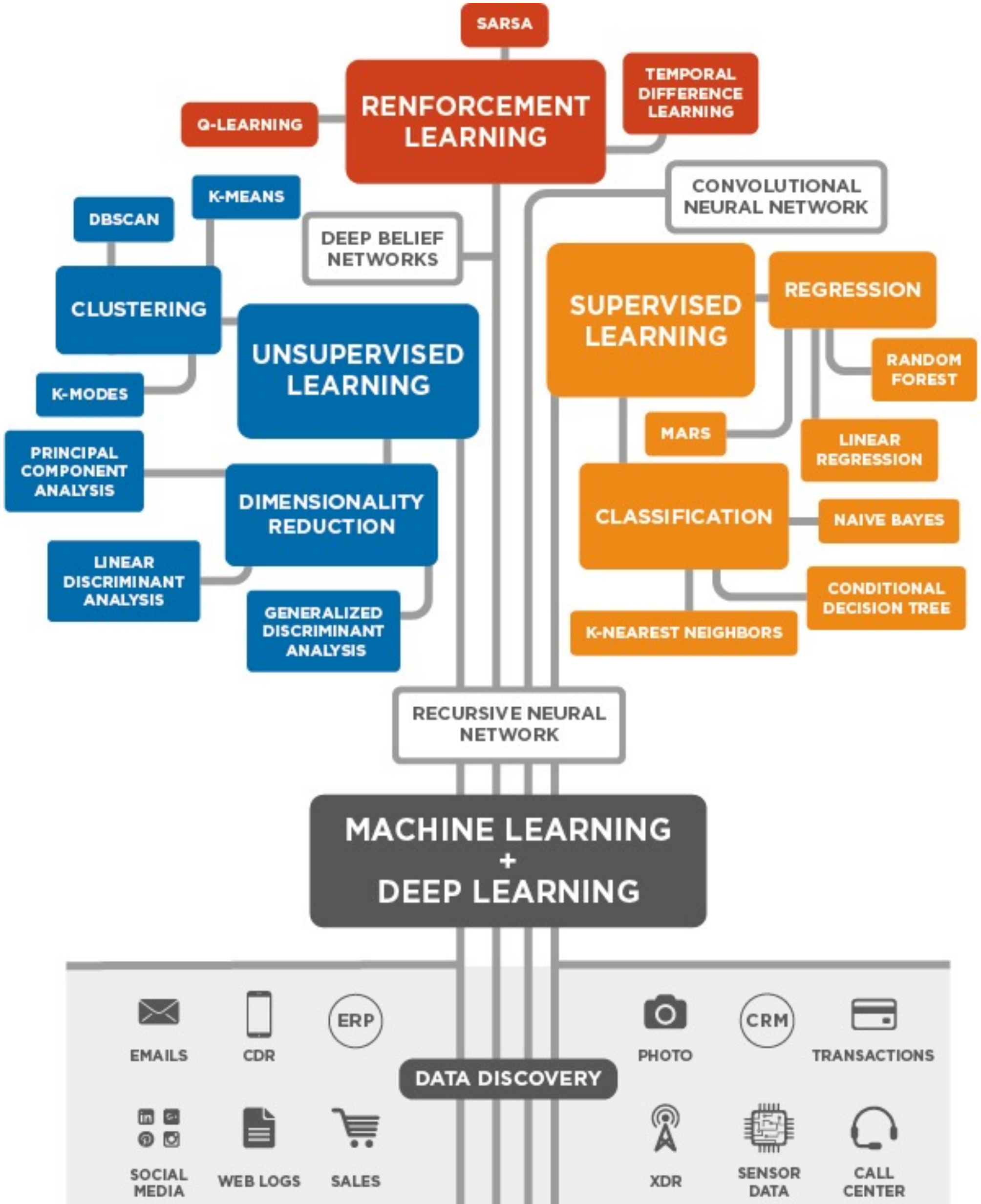
Source: O. Bustos and A. Pomares-Quimbaya (2020), "Stock Market Movement Forecast: A Systematic Review." Expert Systems with Applications (2020): 113464.

AI, ML, DL



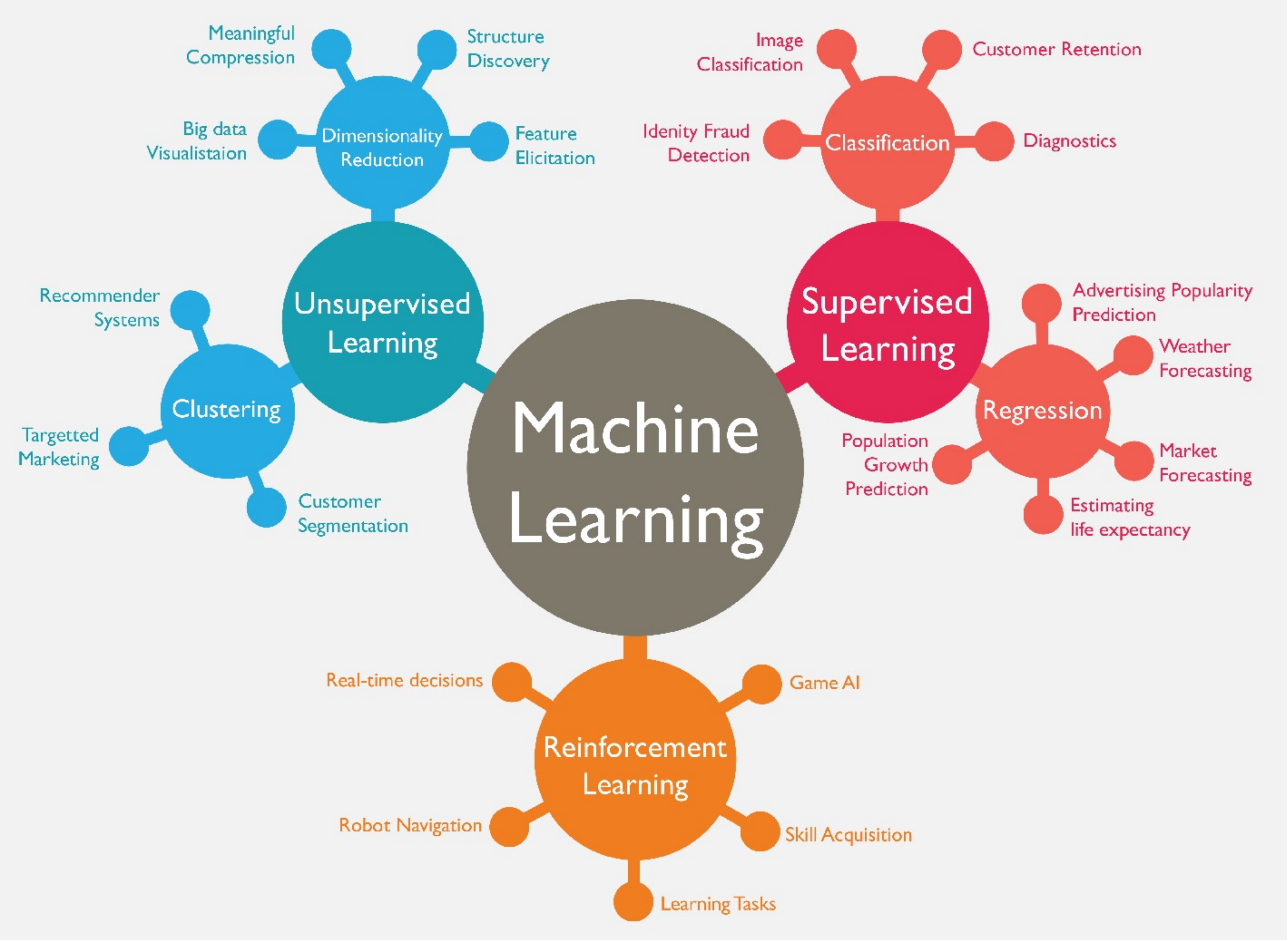
Source: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/deep_learning.html

3 Machine Learning Algorithms



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

Machine Learning (ML)



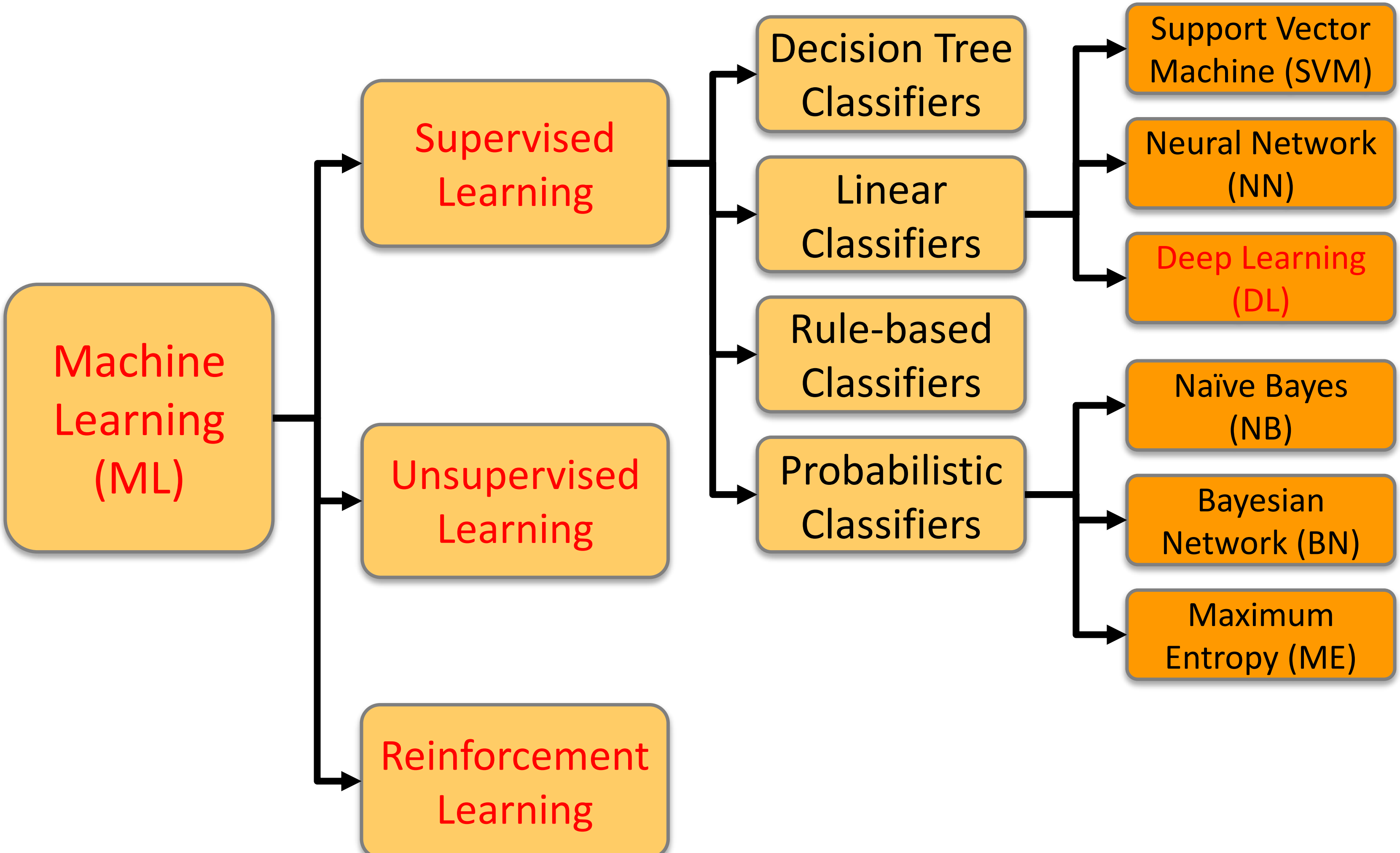
Source: https://www.mactores.com/services/aws-big-data-machine-learning-cognitive-services/

Machine Learning Models

- Deep Learning
- Kernel
- Association rules
- Ensemble
- Decision tree
- Dimensionality reduction
- Clustering
- Regression Analysis
- Bayesian
- Instance based

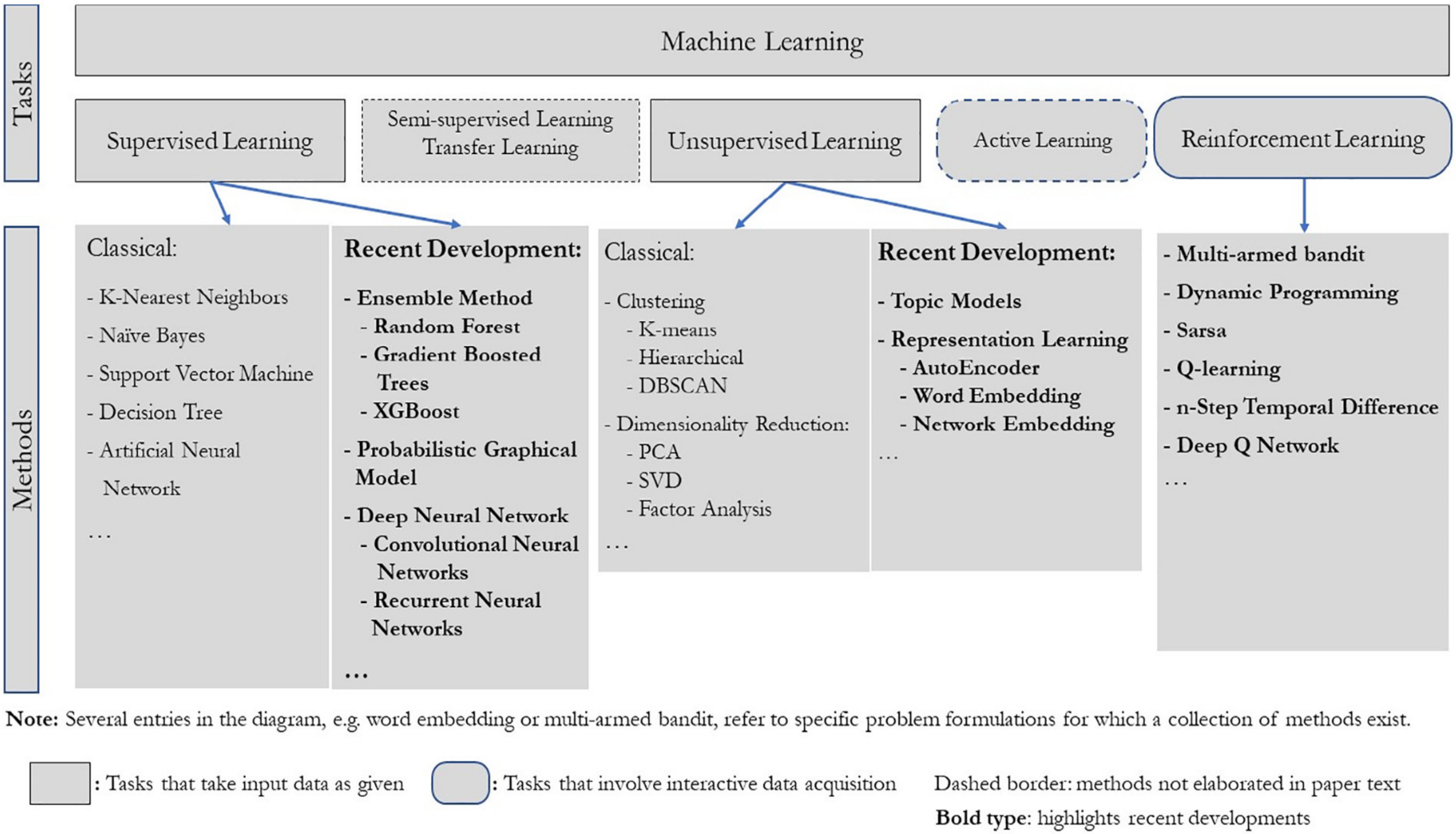
Source: Sunila Gollapudi (2016), Practical Machine Learning, Packt Publishing

Machine Learning (ML) / Deep Learning (DL)



Source: Jesus Serrano-Guerrero, Jose A. Olivas, Francisco P. Romero, and Enrique Herrera-Viedma (2015), "Sentiment analysis: A review and comparative analysis of web services," Information Sciences, 311, pp. 18-38.

Machine Learning Tasks and Methods



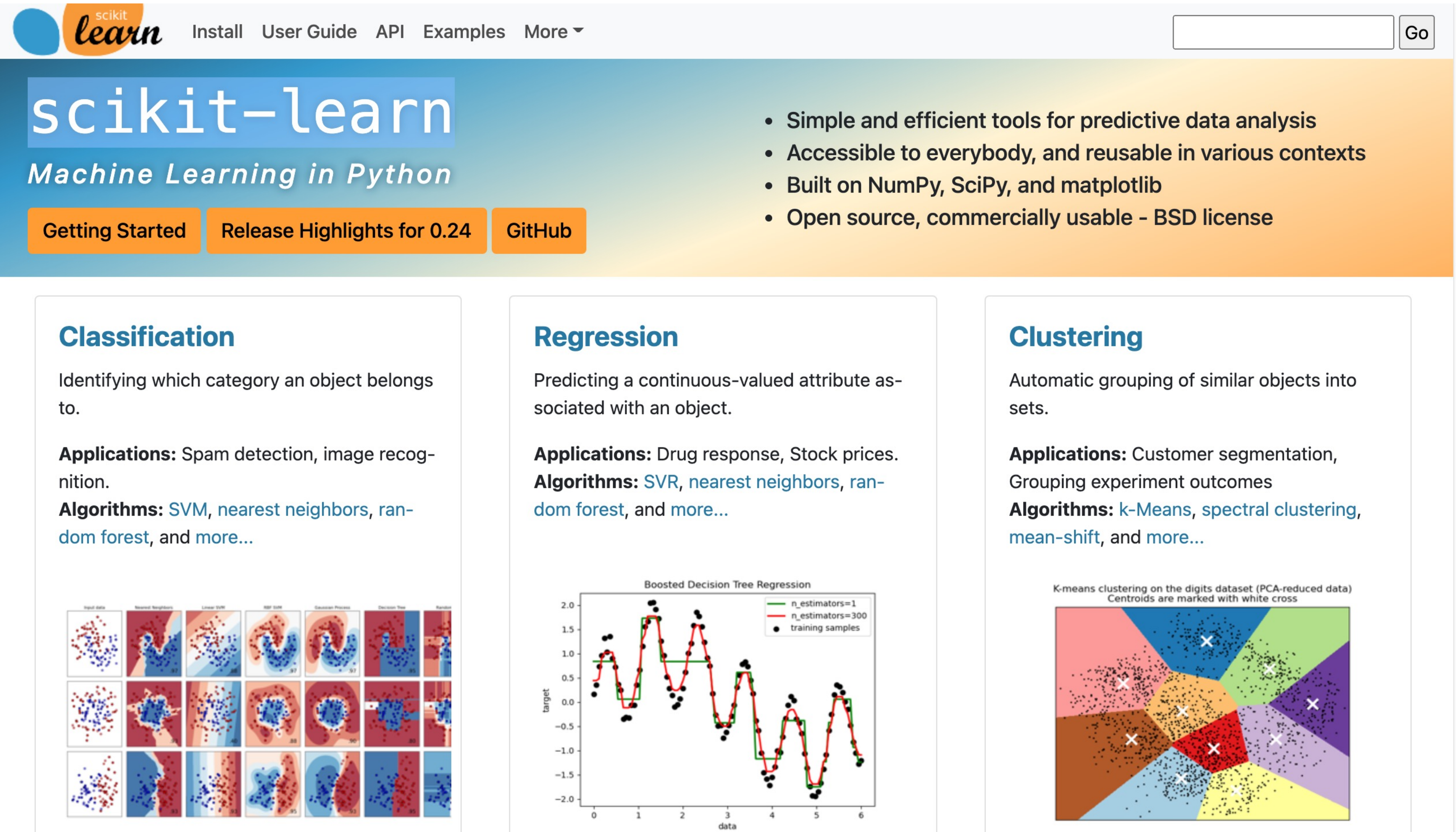
Source: Liye Ma and Baohong Sun (2020), "Machine learning and AI in marketing – Connecting computing power to human insights." International Journal of Research in Marketing, 37, no. 3, 481-504.

Machine Learning

Scikit-Learn

Machine Learning in Python

Scikit-Learn



Source: http://scikit-learn.org/

Machine Learning
Supervised Learning (Classification)
Learning from Examples

$$y = f(x)$$

x	y

Machine Learning
Supervised Learning (Classification)
Learning from Examples

$$y = f(x)$$

x

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

y

Linear function

$$y = f(x)$$

$$y = w_1 x + w_0$$

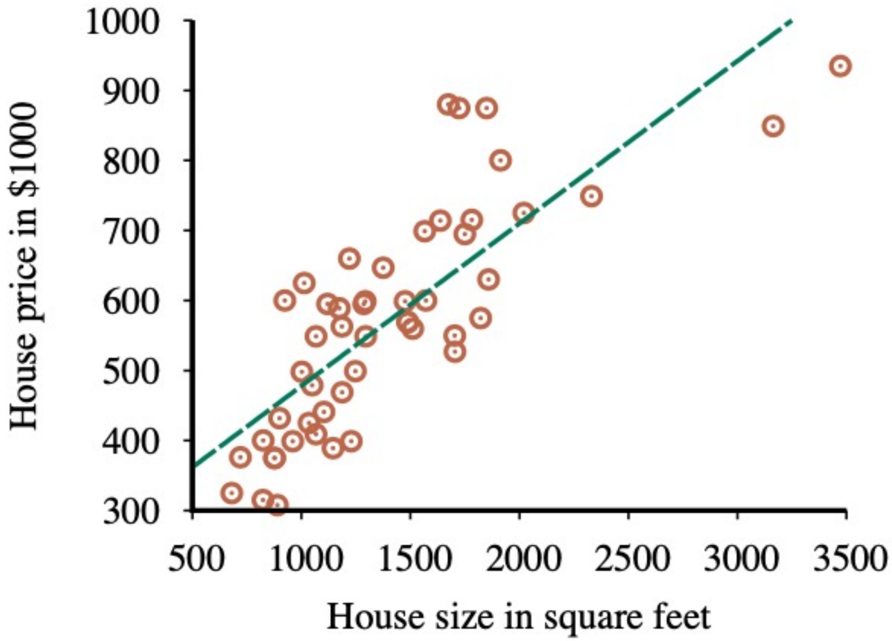
$$h_w(x) = w_1 x + w_0$$

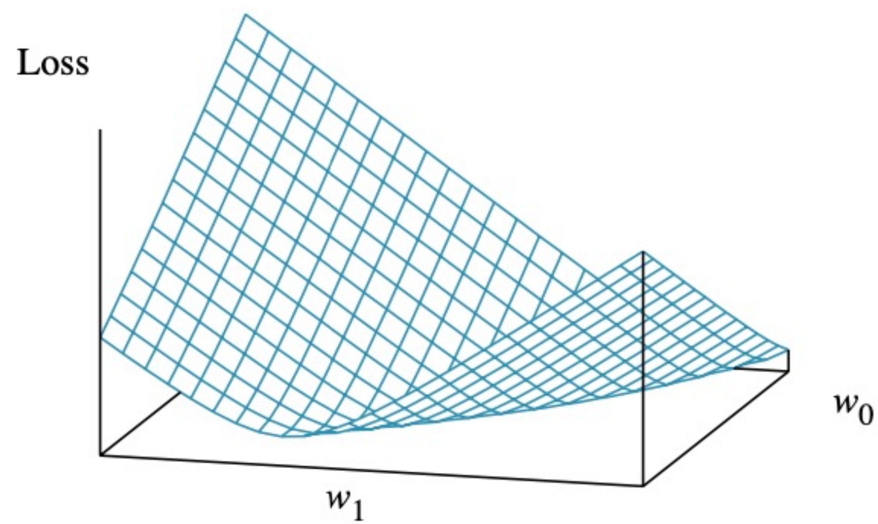
Source: Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson

Linear Regression Weight Space

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

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





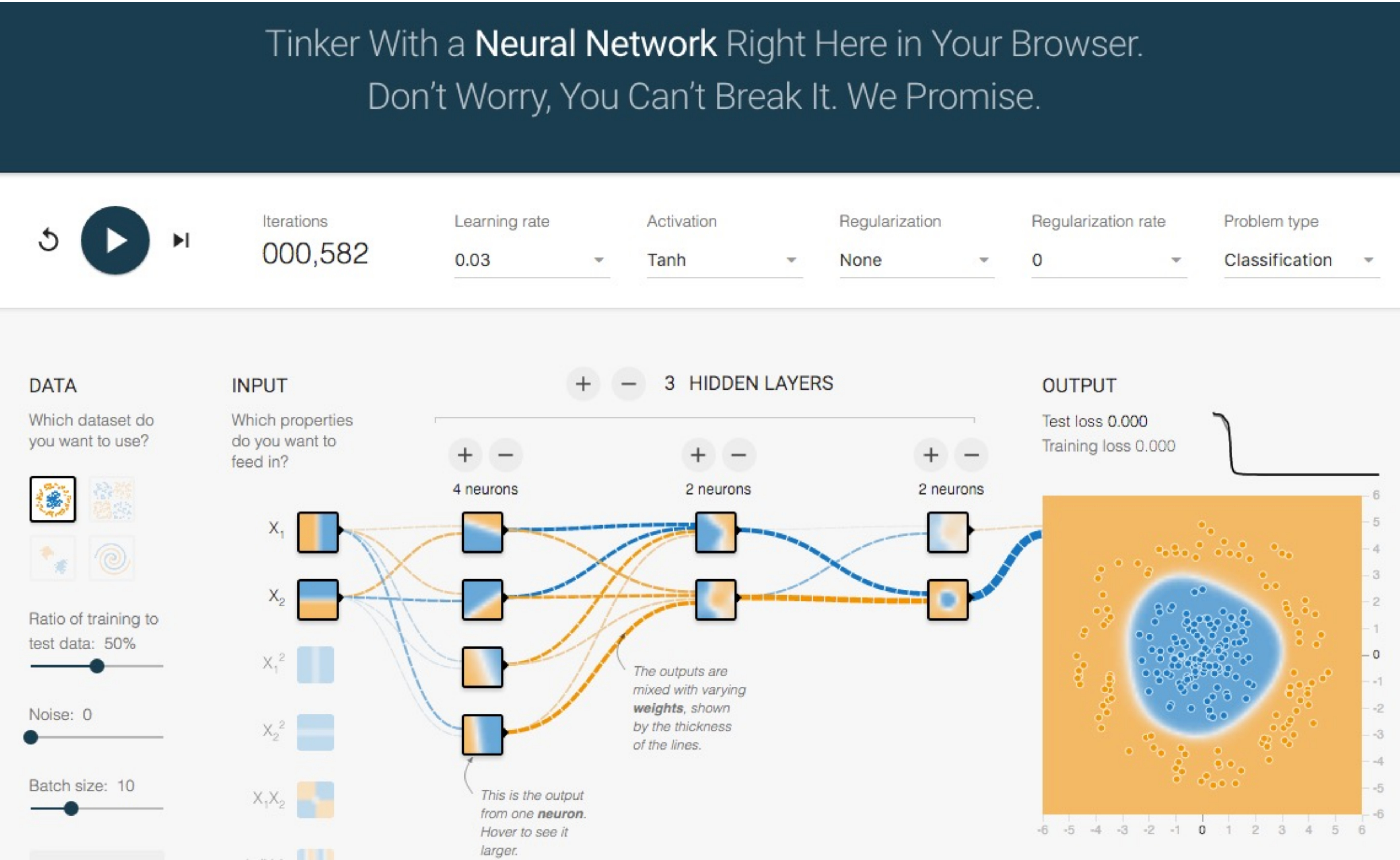
$y = 0.232\, x + 246$

Loss function for Weights (w_1, w_0)

Deep Learning

Deep Learning
and
Neural Networks

TensorFlow Playground



http://playground.tensorflow.org/

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

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

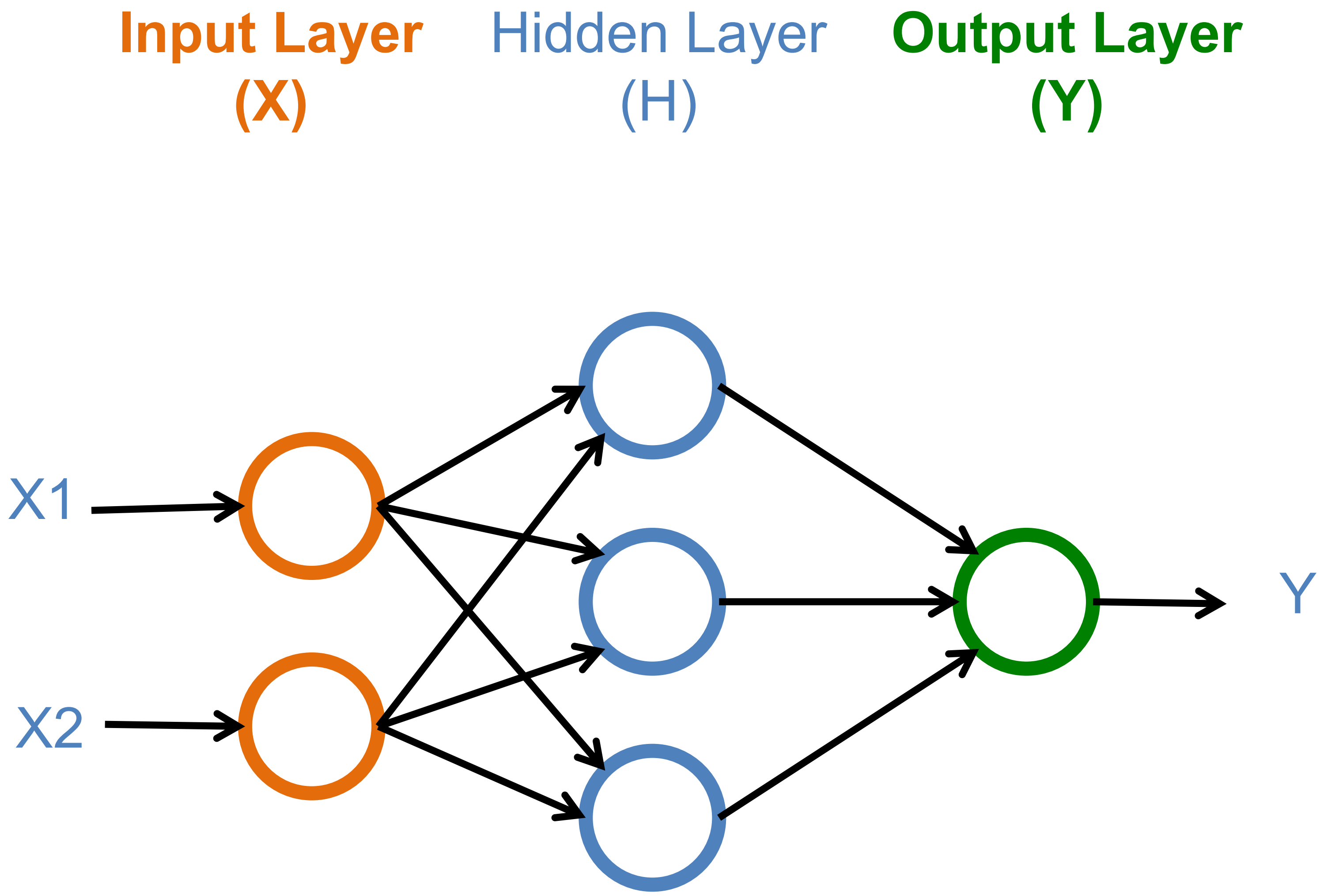
Tensor

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

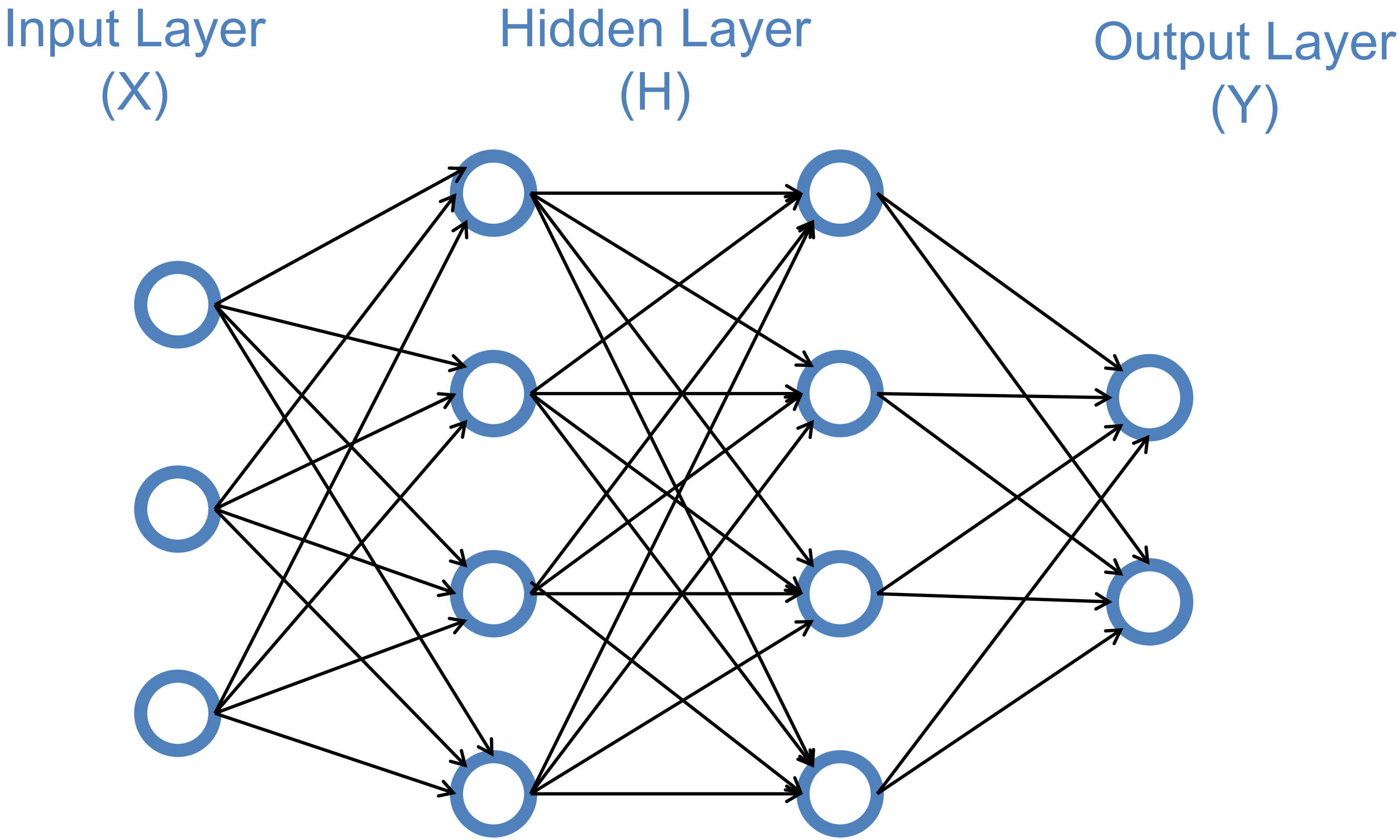
Deep Learning
and
Neural Networks

Deep Learning Foundations: Neural Networks

Deep Learning and Neural Networks



Deep Learning and Neural Networks



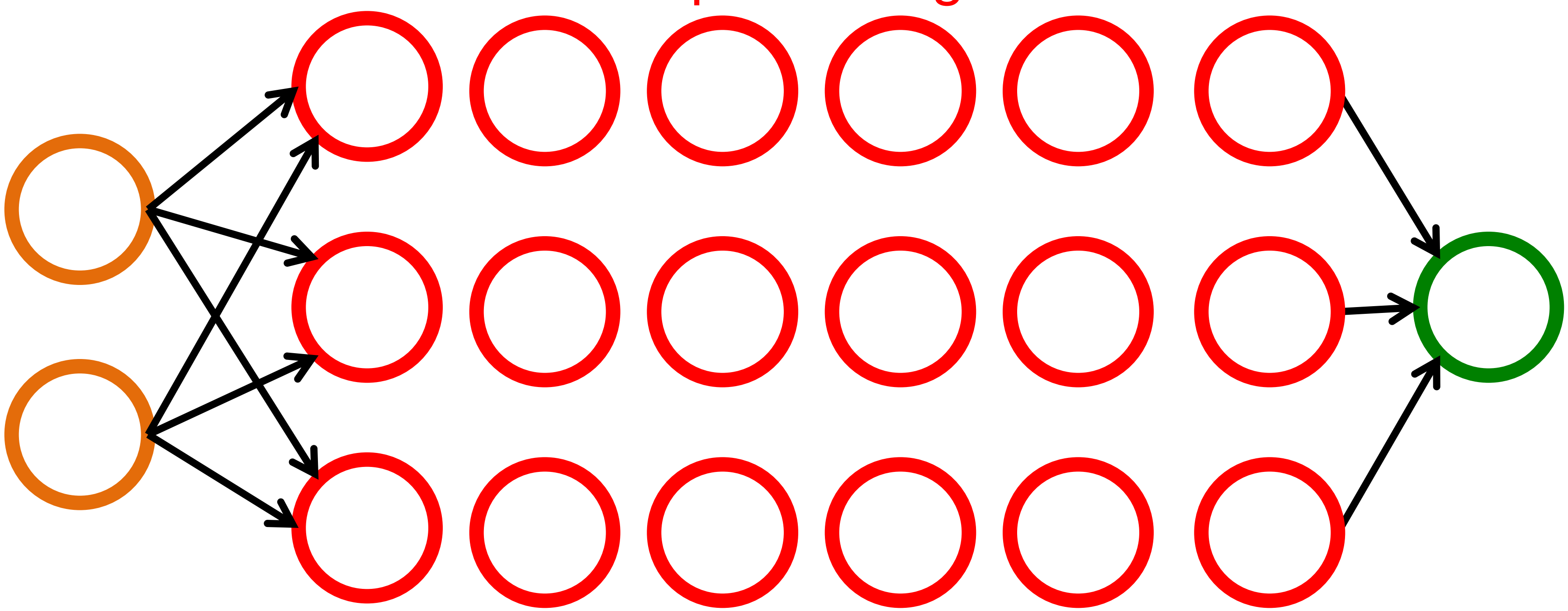
Deep Learning and Neural Networks

Input Layer
(X)

Hidden Layers
(H)

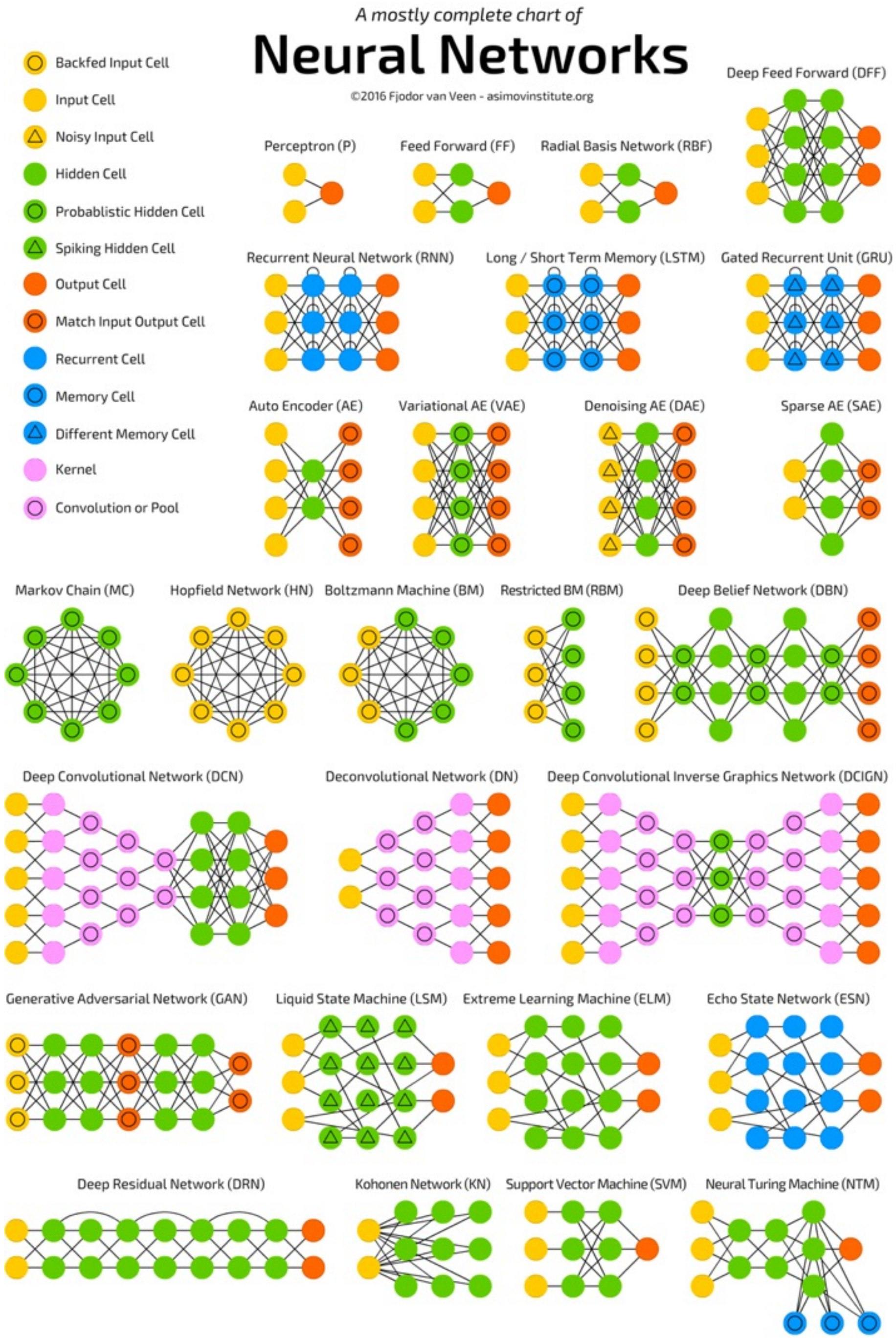
Output Layer
(Y)

Deep Neural Networks
Deep Learning

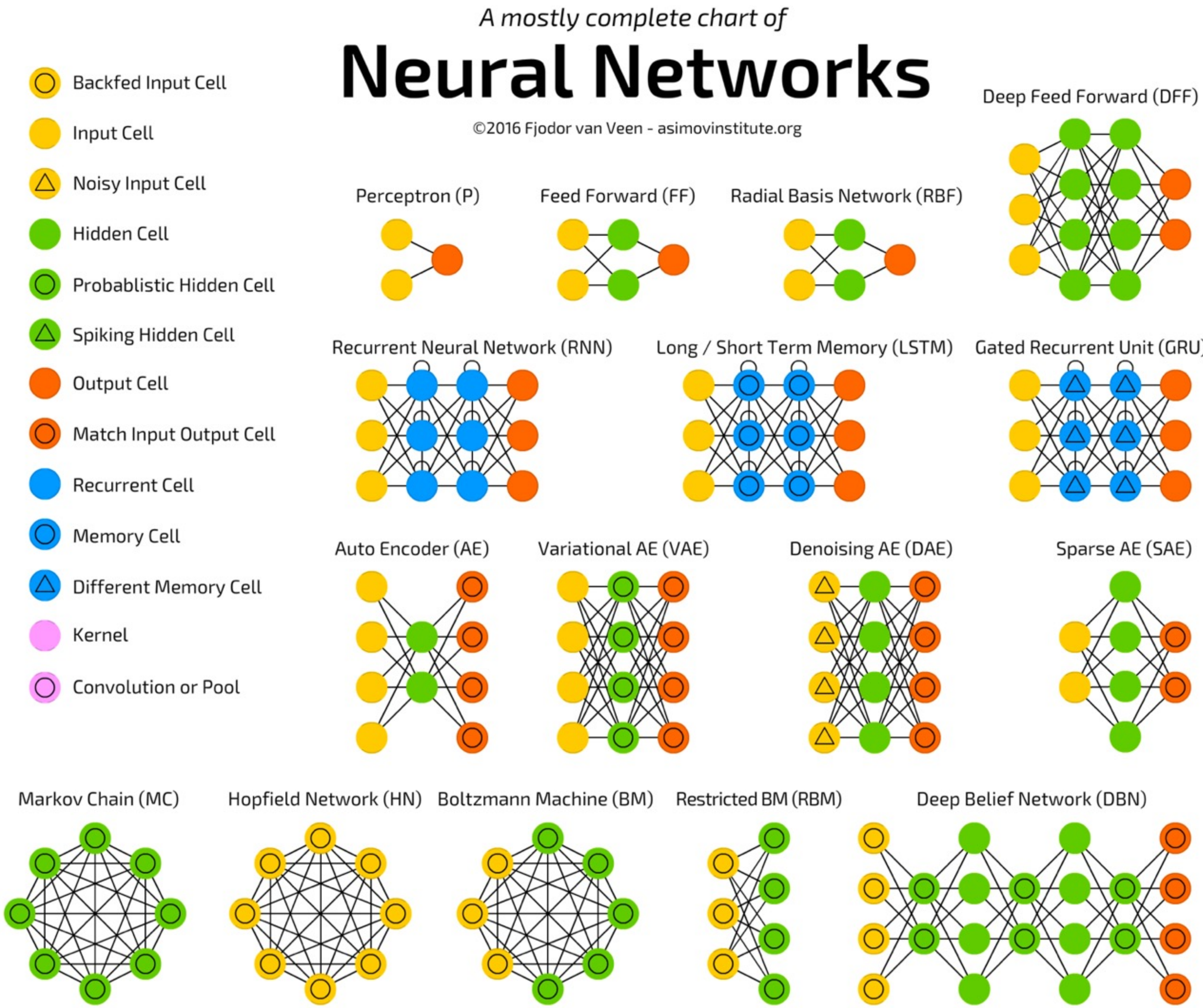


Deep Learning
and
Deep Neural Networks

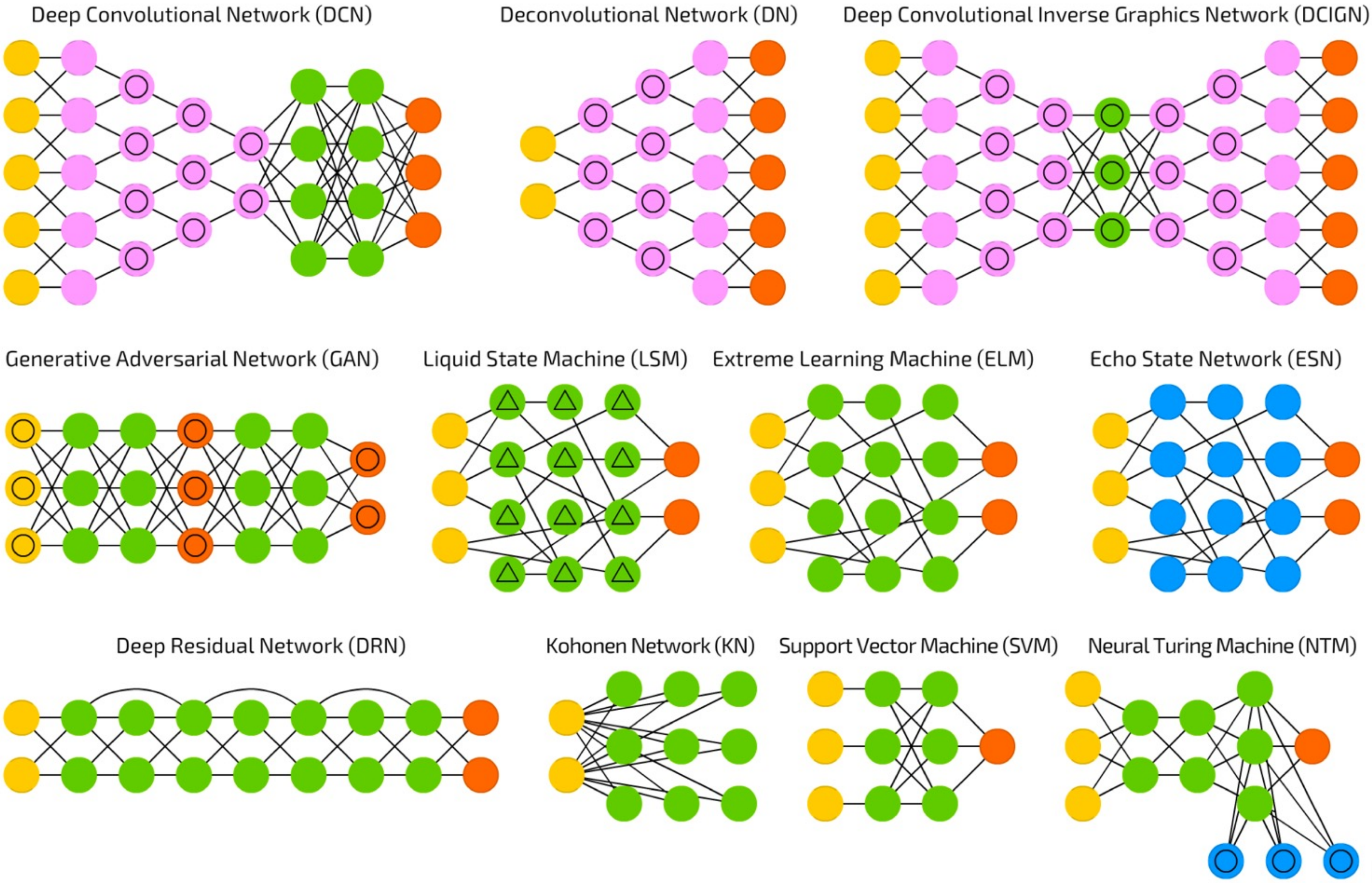
Neural Networks (NN)



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



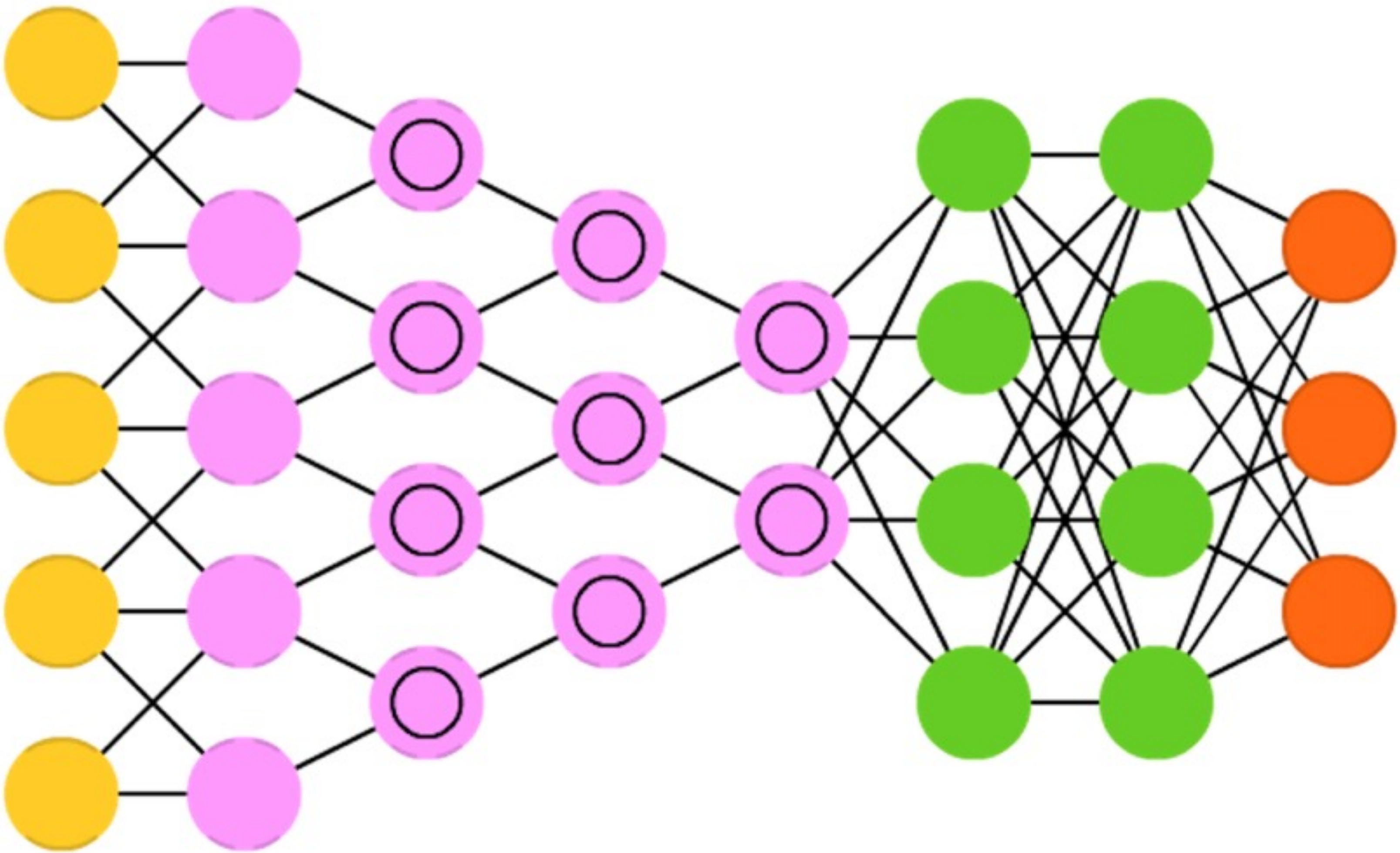
Source: http://www.asimovinstitute.org/neural-network-zoo/



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

Convolutional Neural Networks

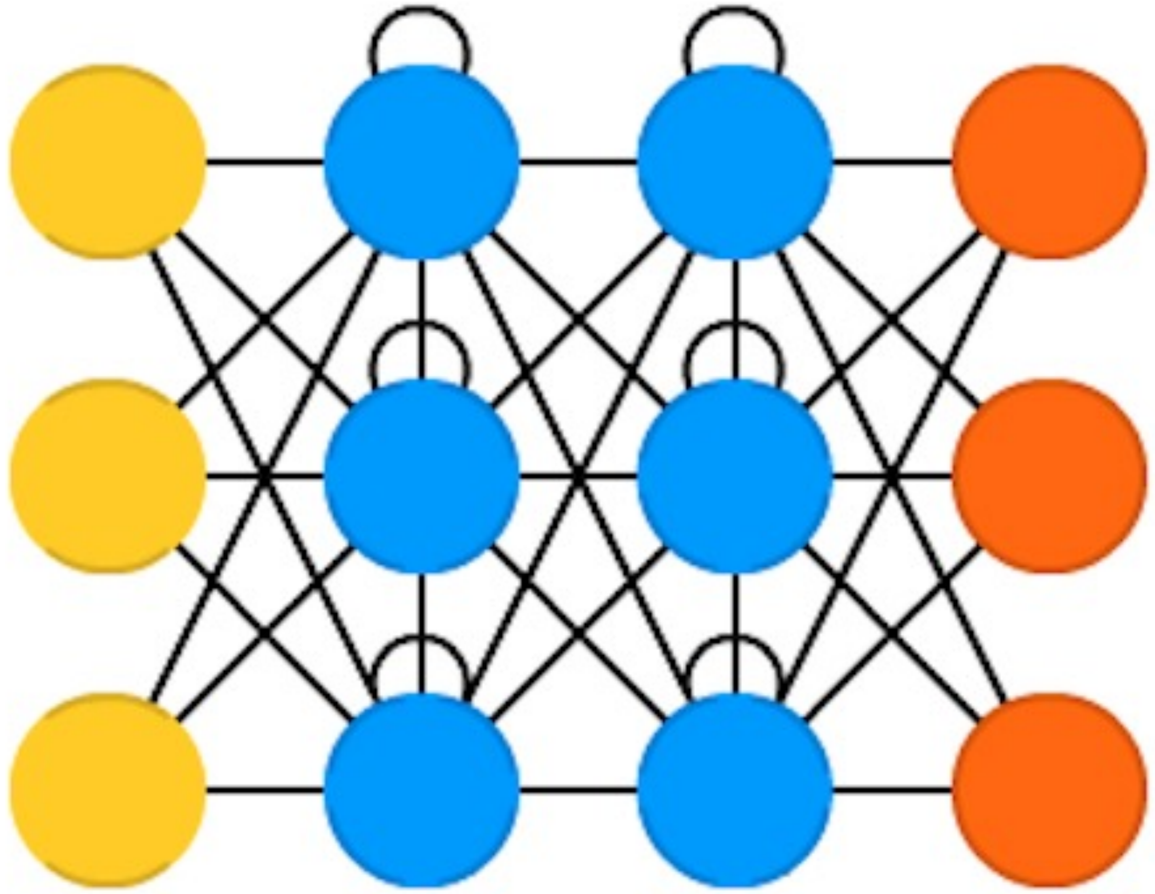
(CNN or Deep Convolutional Neural Networks, DCNN)



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

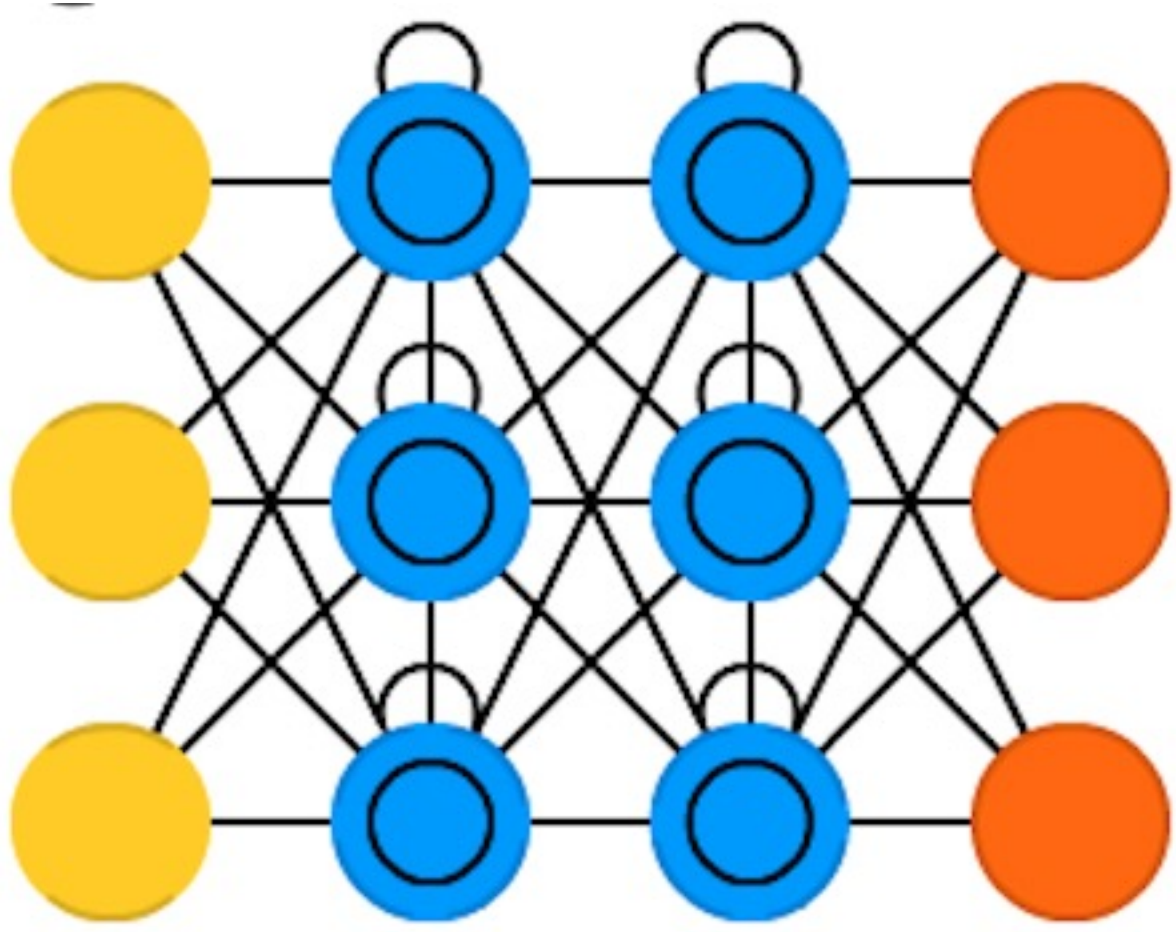
Source: http://www.asimovinstitute.org/neural-network-zoo/

Recurrent Neural Networks (RNN)



Elman, Jeffrey L. "Finding structure in time." Cognitive science 14.2 (1990): 179-211
Source: http://www.asimovinstitute.org/neural-network-zoo/

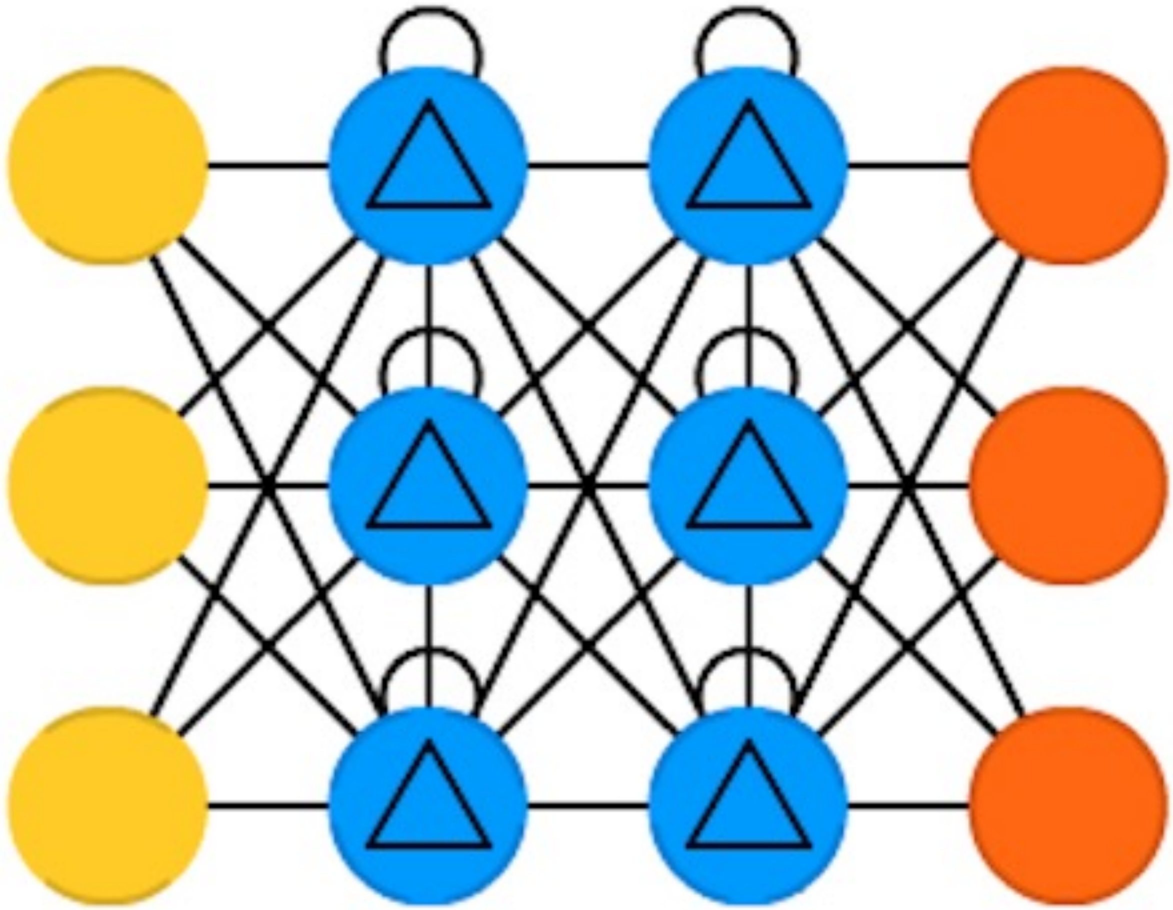
Long / Short Term Memory (LSTM)



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

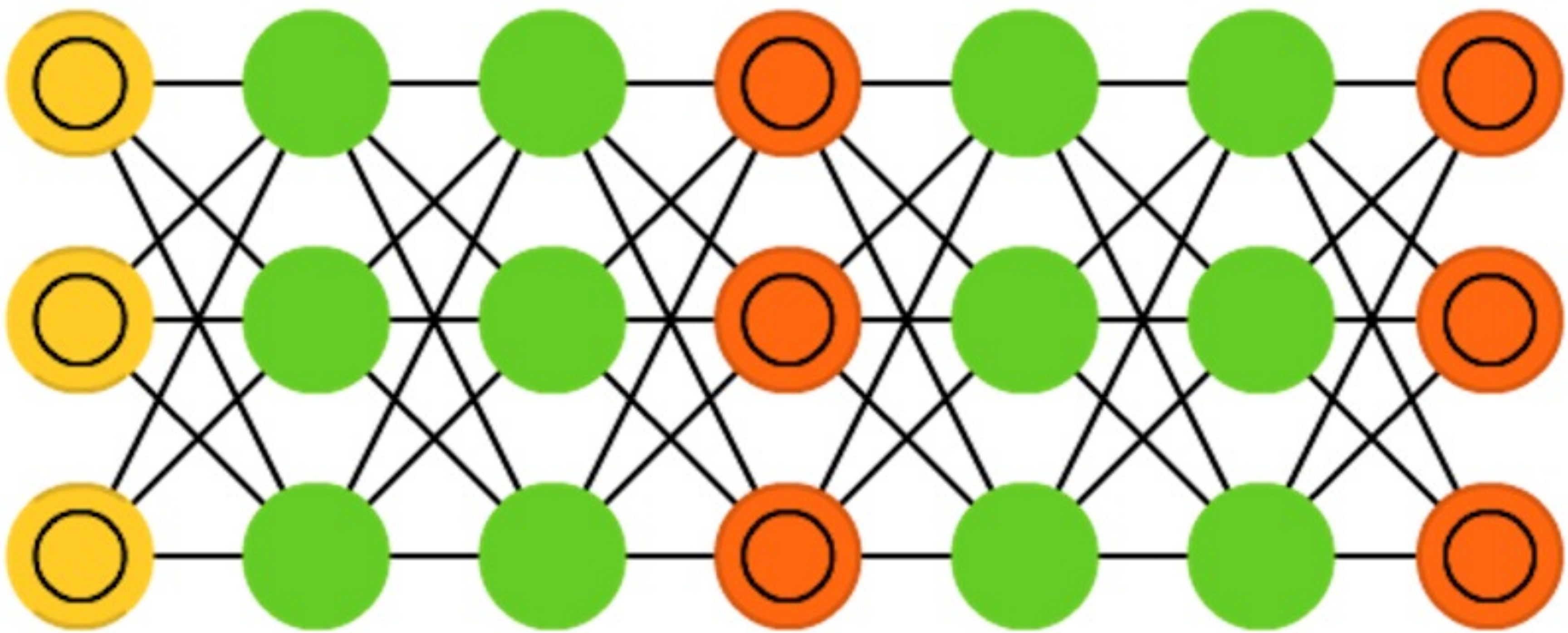
Source: http://www.asimovinstitute.org/neural-network-zoo/

Gated Recurrent Units (GRU)



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

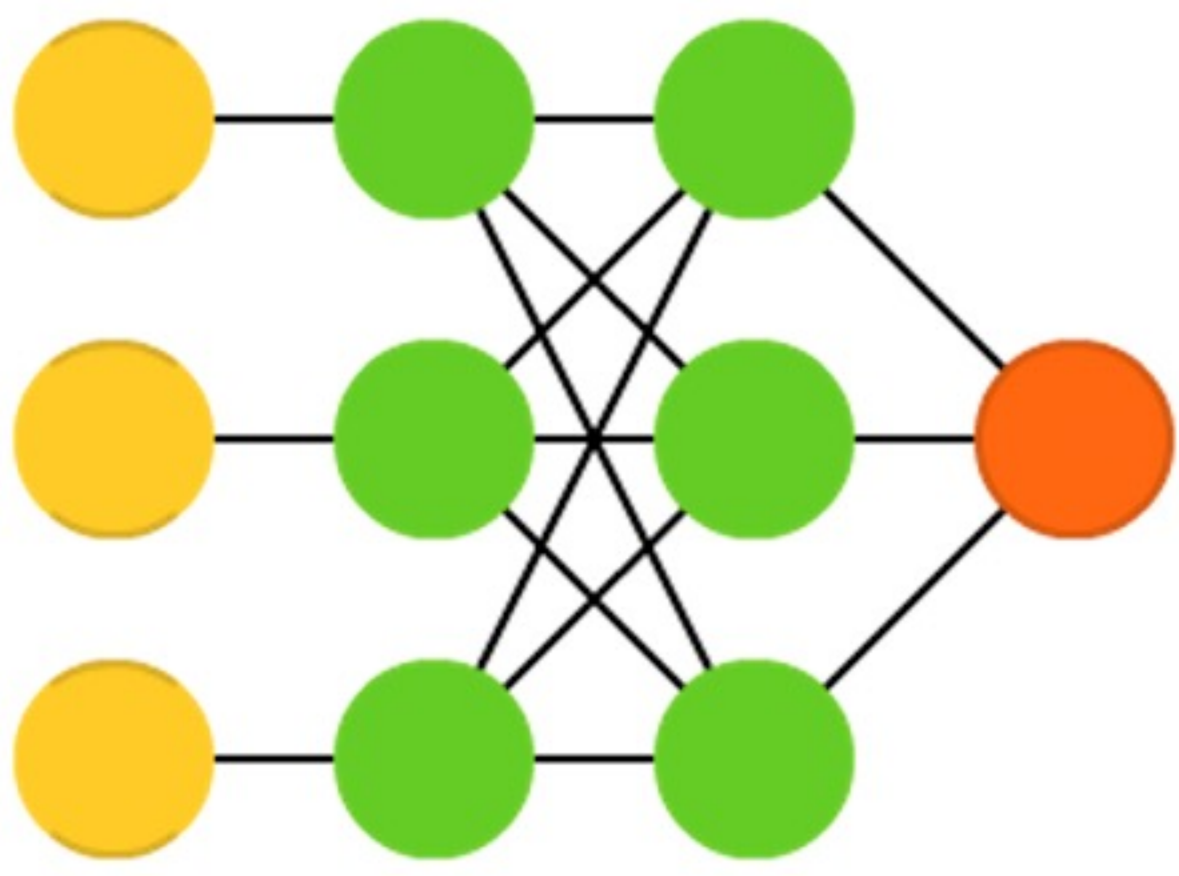
Generative Adversarial Networks (GAN)



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

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

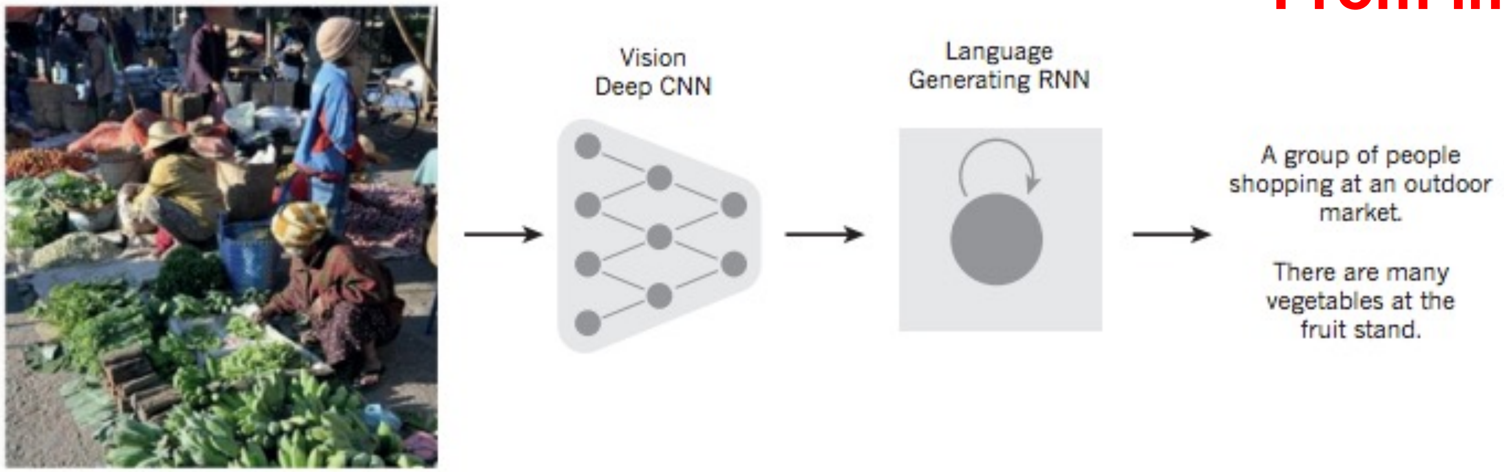
Support Vector Machines (SVM)



Cortes, Corinna, and Vladimir Vapnik. “Support-vector networks.” Machine learning 20.3 (1995): 273-297.

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

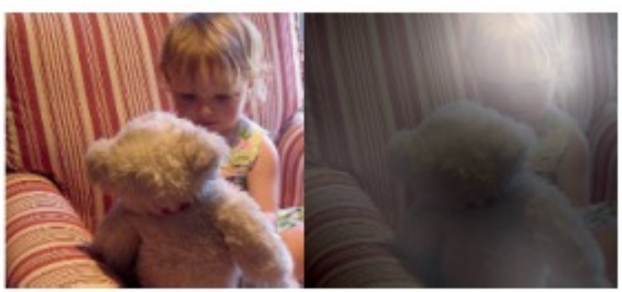
From image to text



A woman is throwing a **frisbee** in a park.

A **dog** is standing on a hardwood floor.

A **stop** sign is on a road with a mountain in the background



A little **girl** sitting on a bed with a teddy bear.

A group of **people** sitting on a boat in the water.

A giraffe standing in a forest with **trees** in the background.

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

From image to text

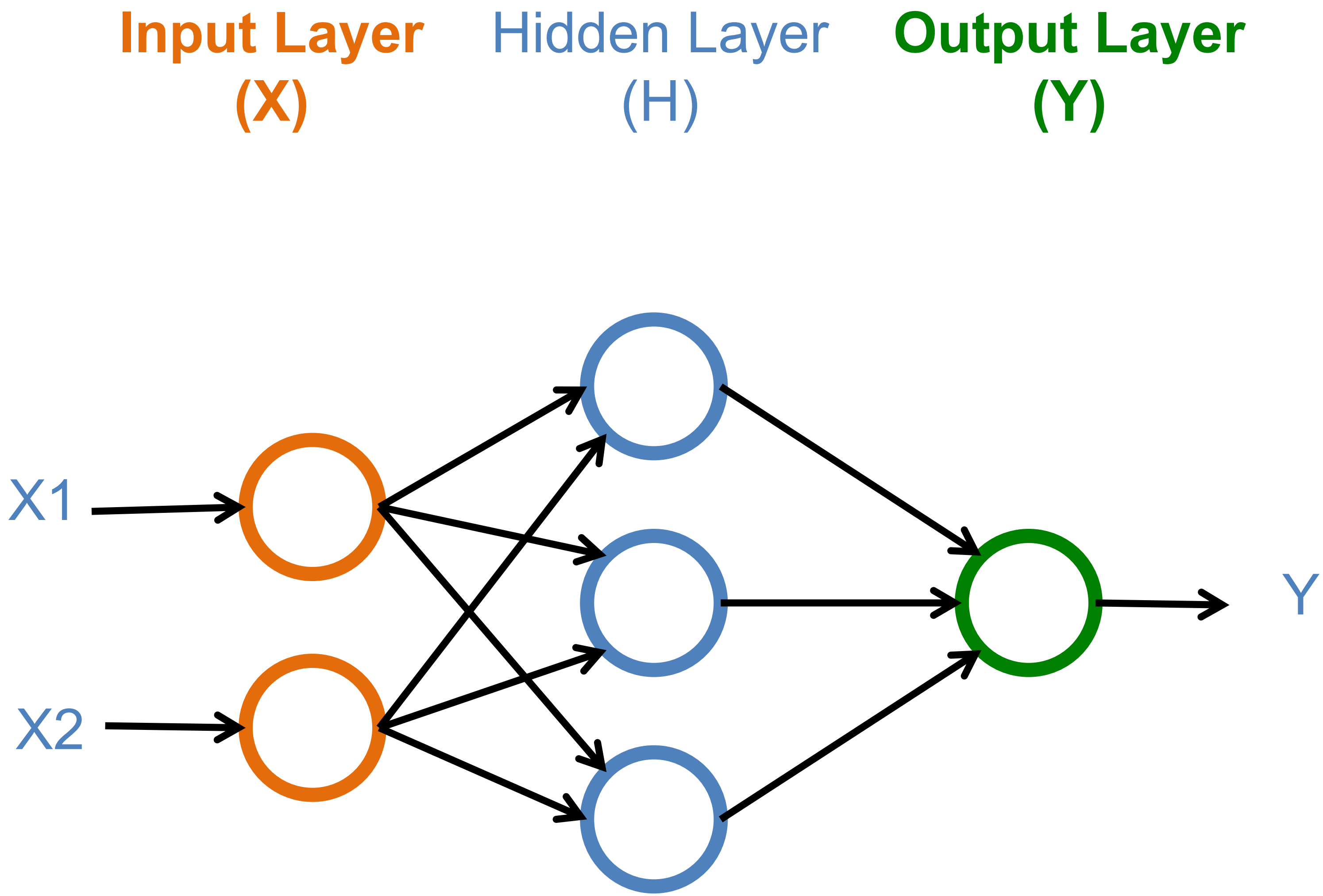
Image: deep convolution neural network (CNN)

Text: recurrent neural network (RNN)

A group of **people** sitting on a boat in the water.

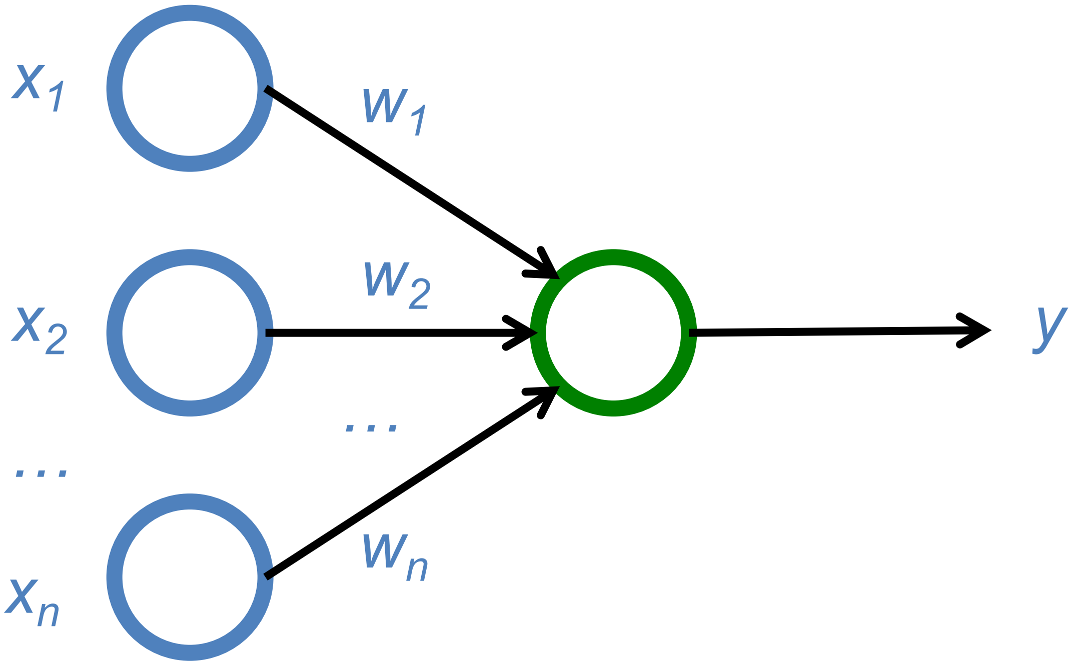
Source: LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521, no. 7553 (2015): 436-444.

Neural Networks



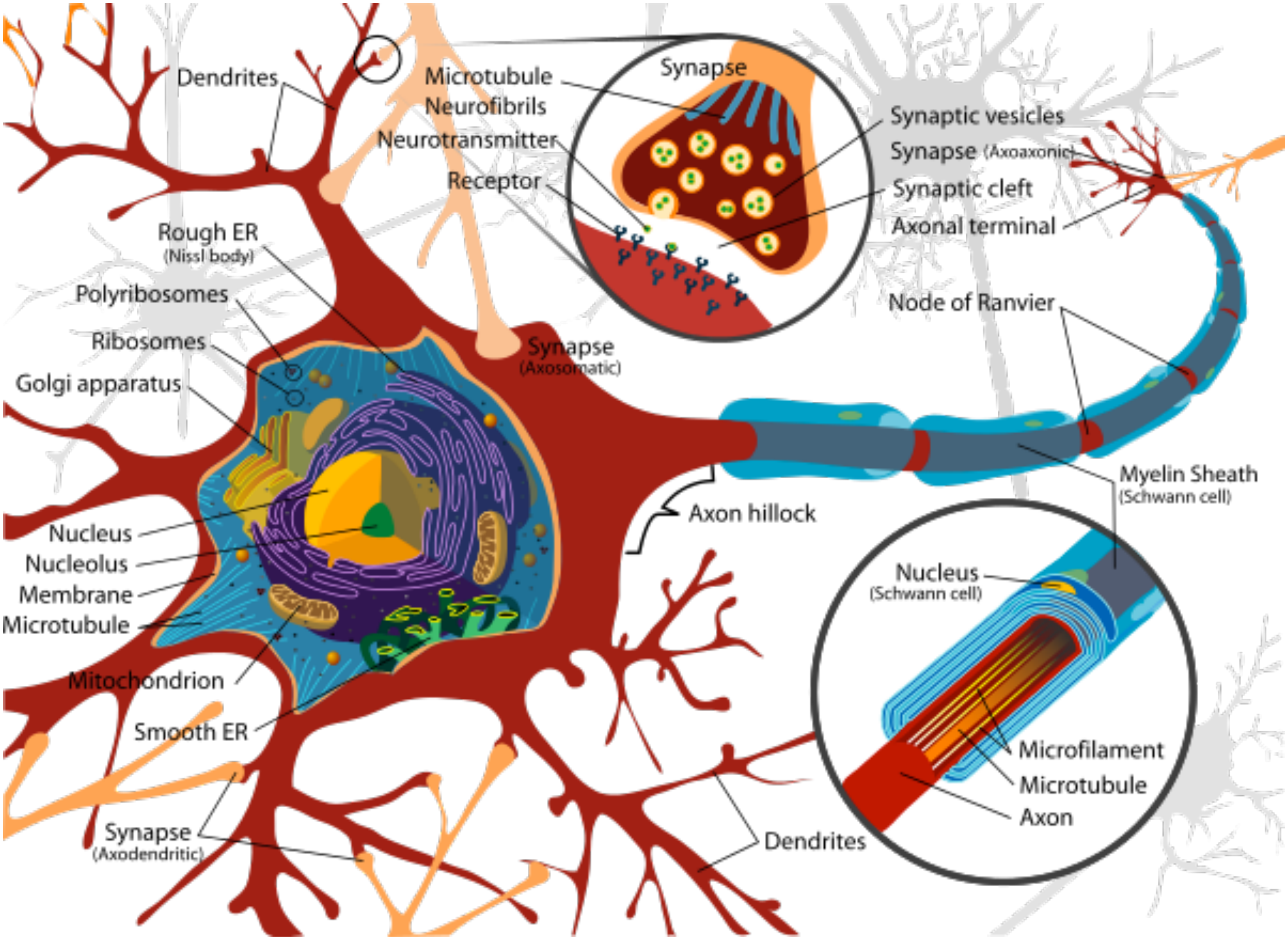
Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

The Neuron



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

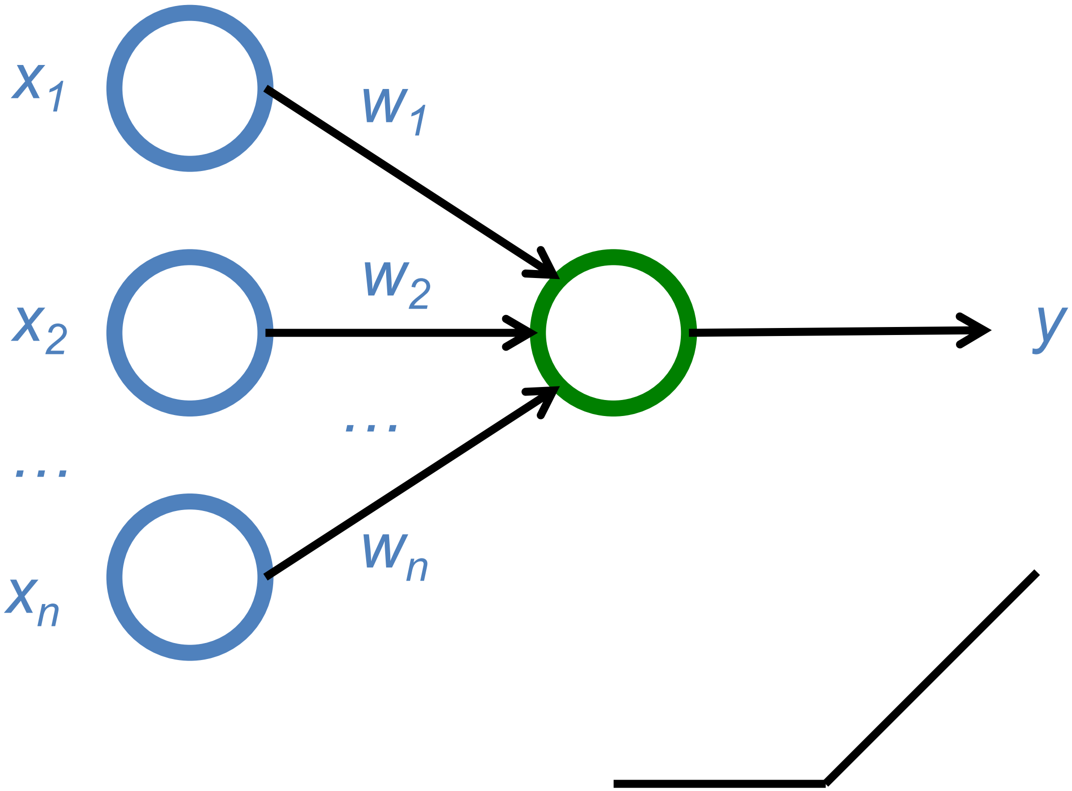
Neuron and Synapse



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

The Neuron

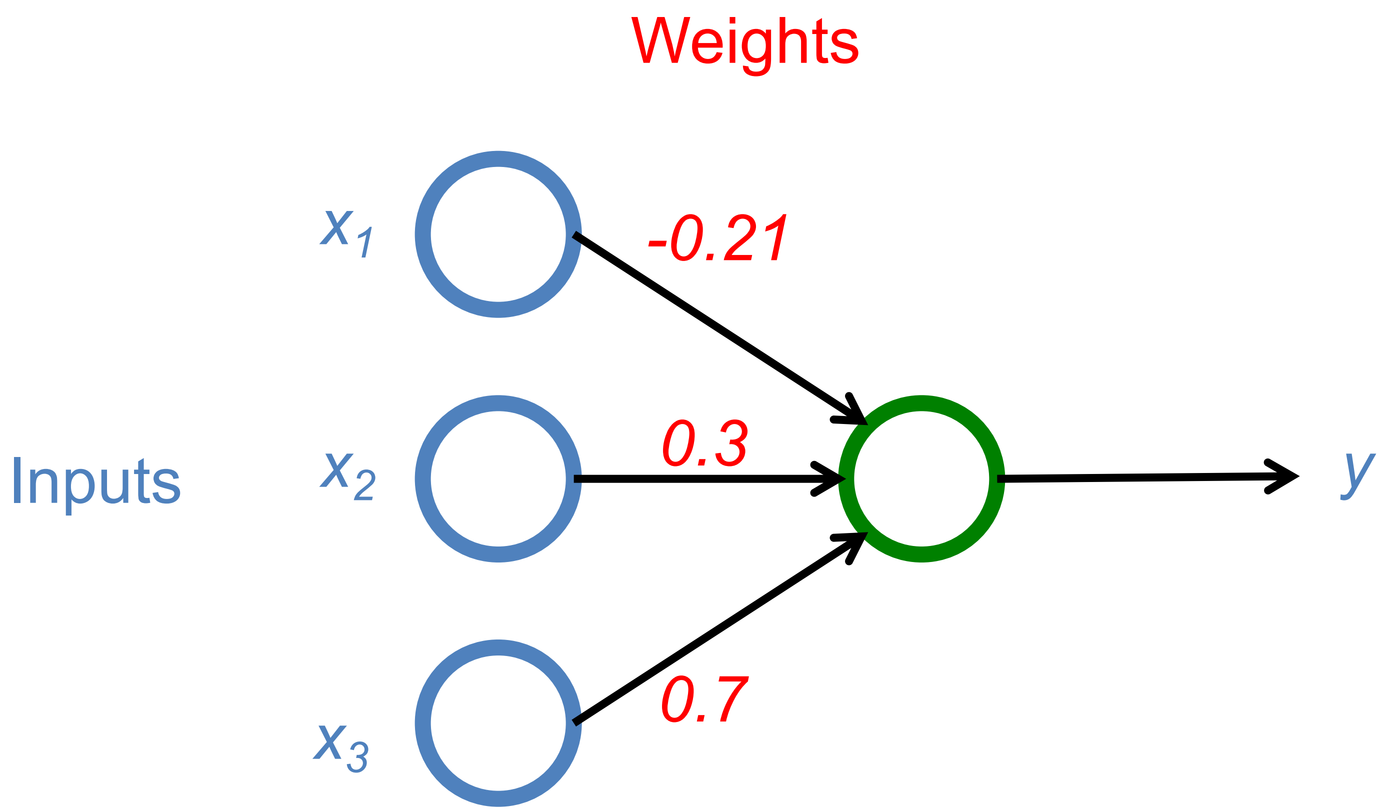
$$y = F\left(\sum_i w_i x_i\right)$$



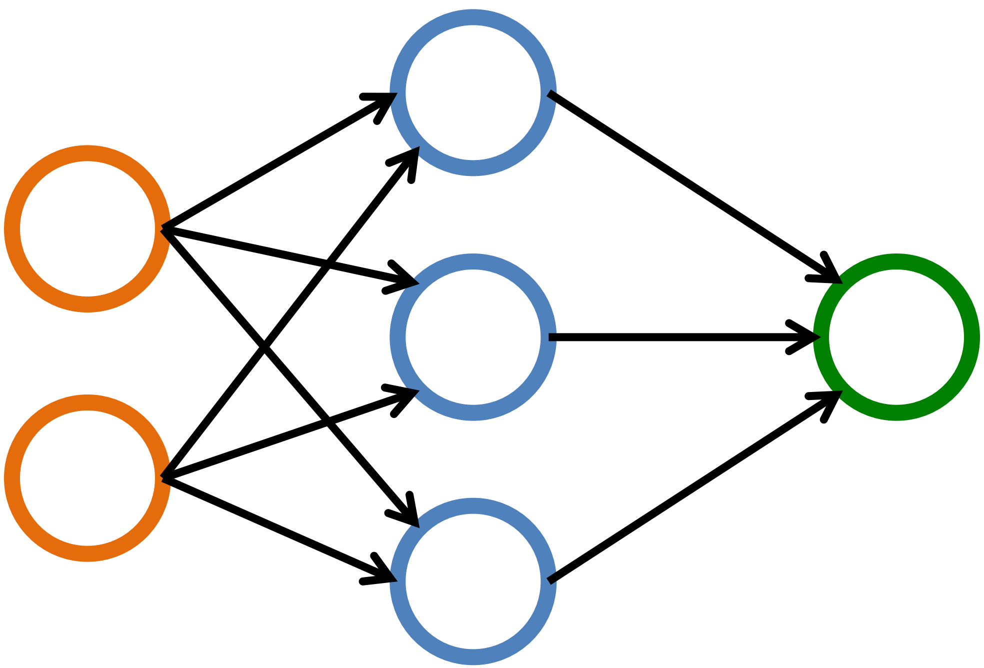
$$F(x) = \max(0, x)$$

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

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

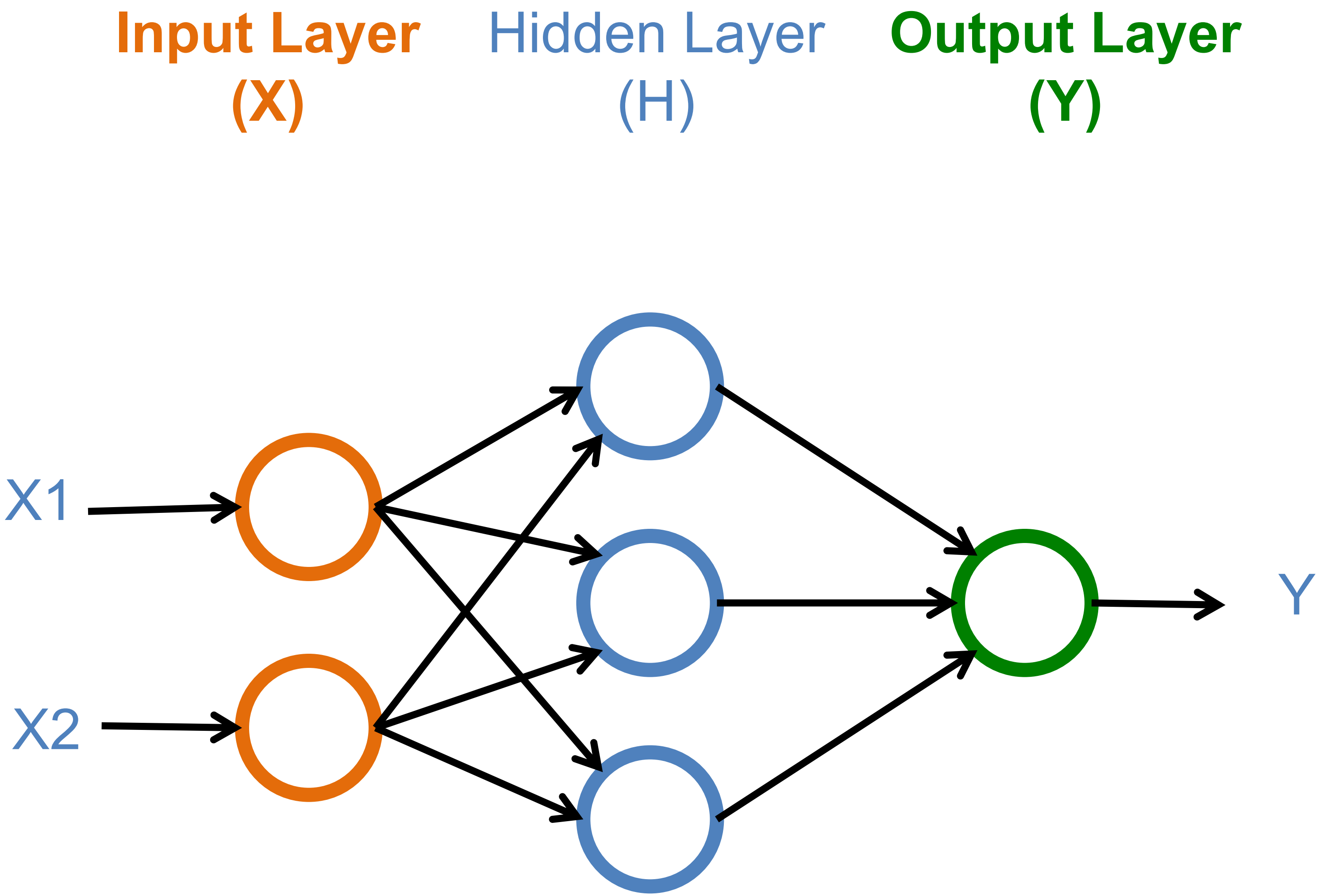


Neural Networks



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

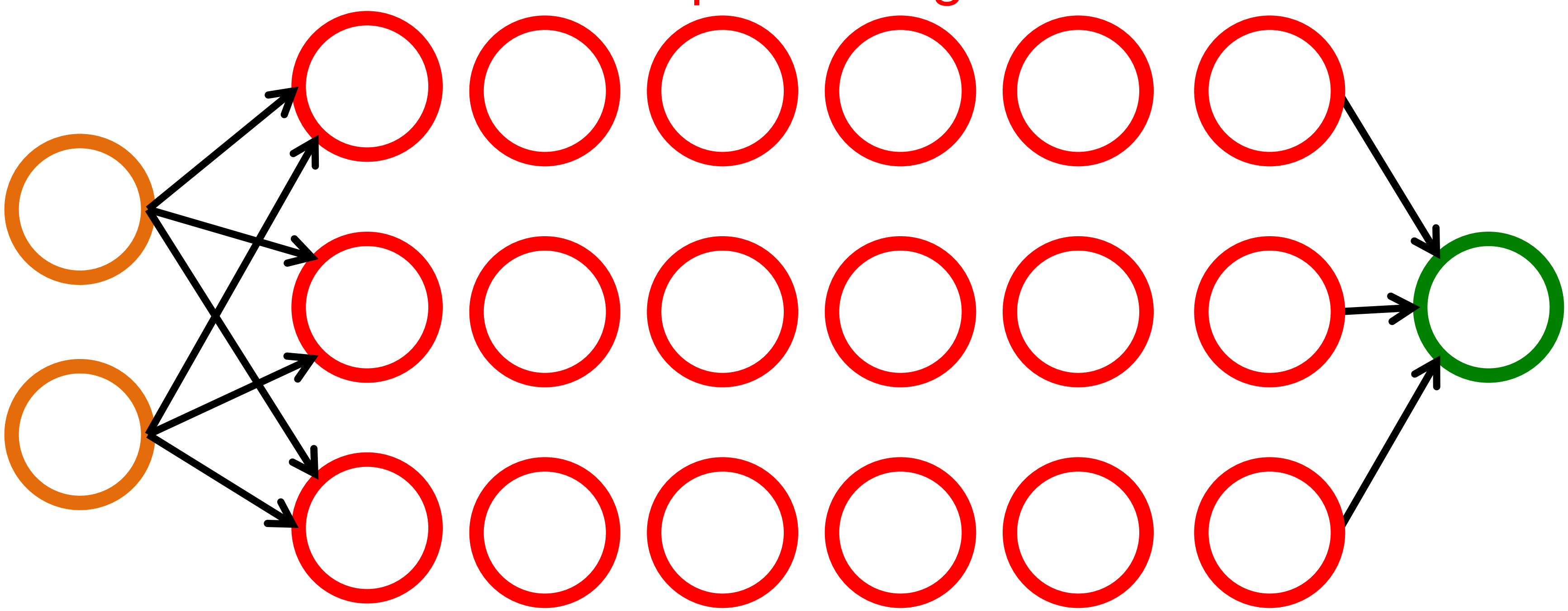
Neural Networks

Input Layer
(X)

Hidden Layers
(H)

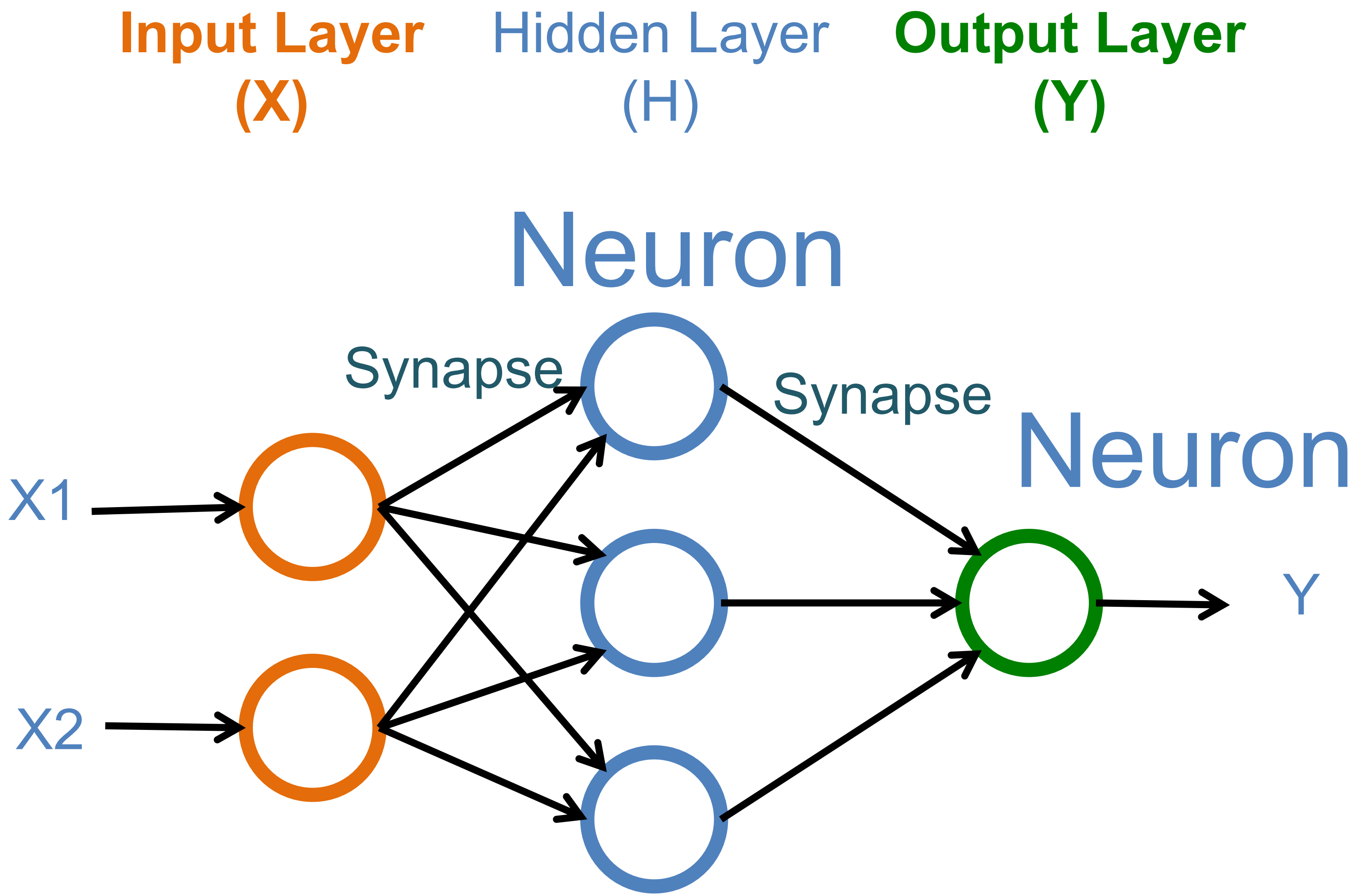
Output Layer
(Y)

Deep Neural Networks
Deep Learning



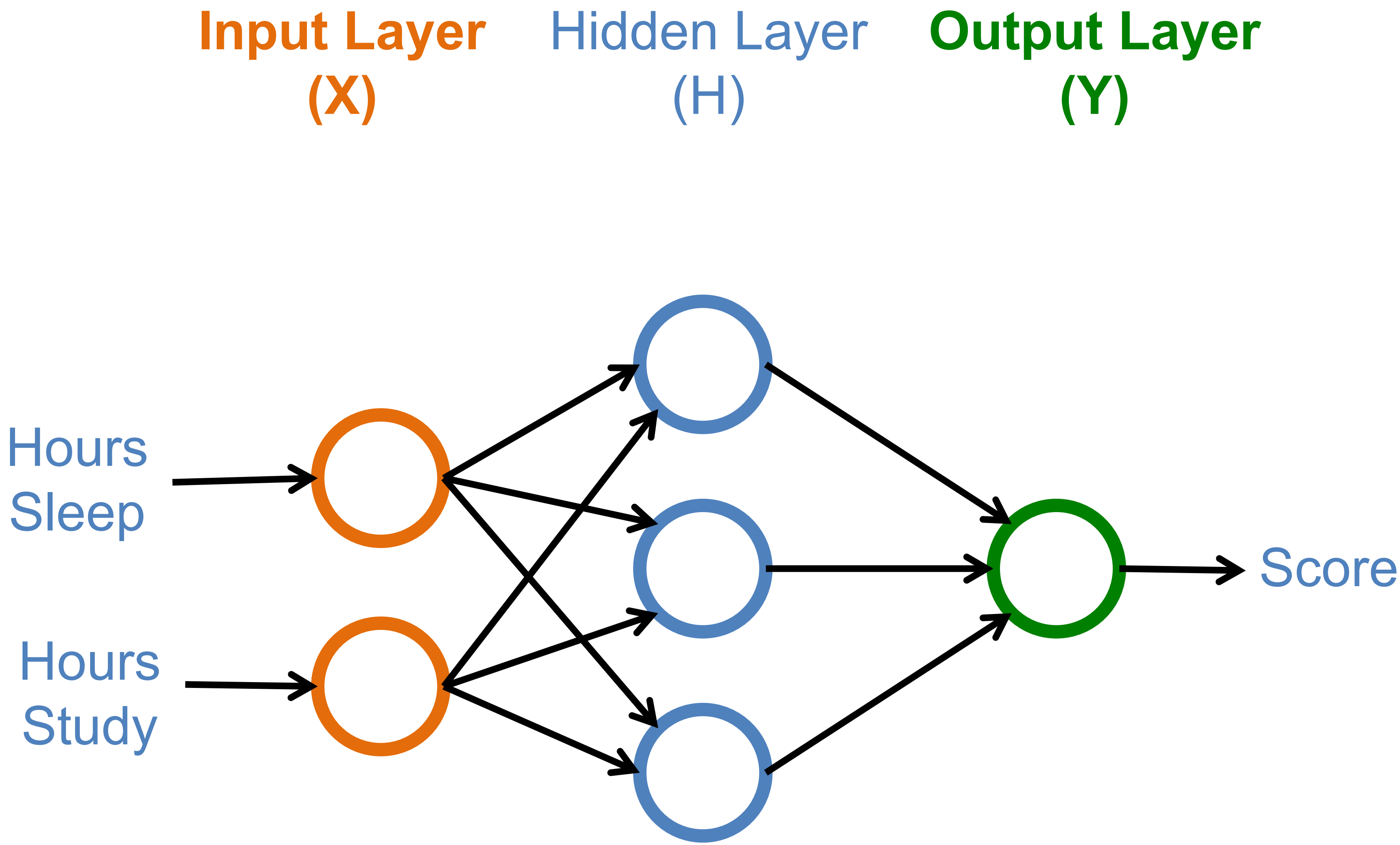
Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Neural Networks



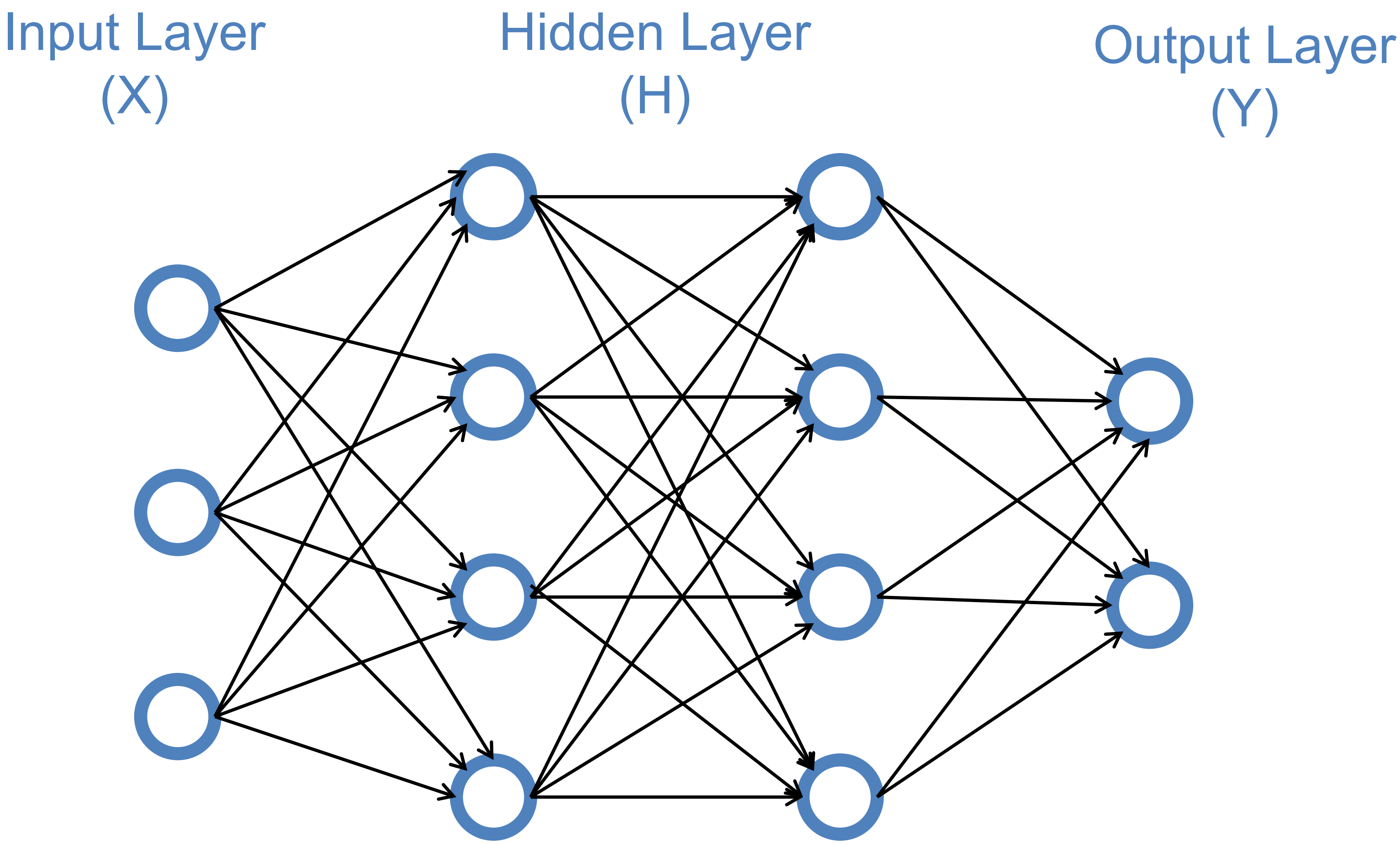
Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Neural Networks



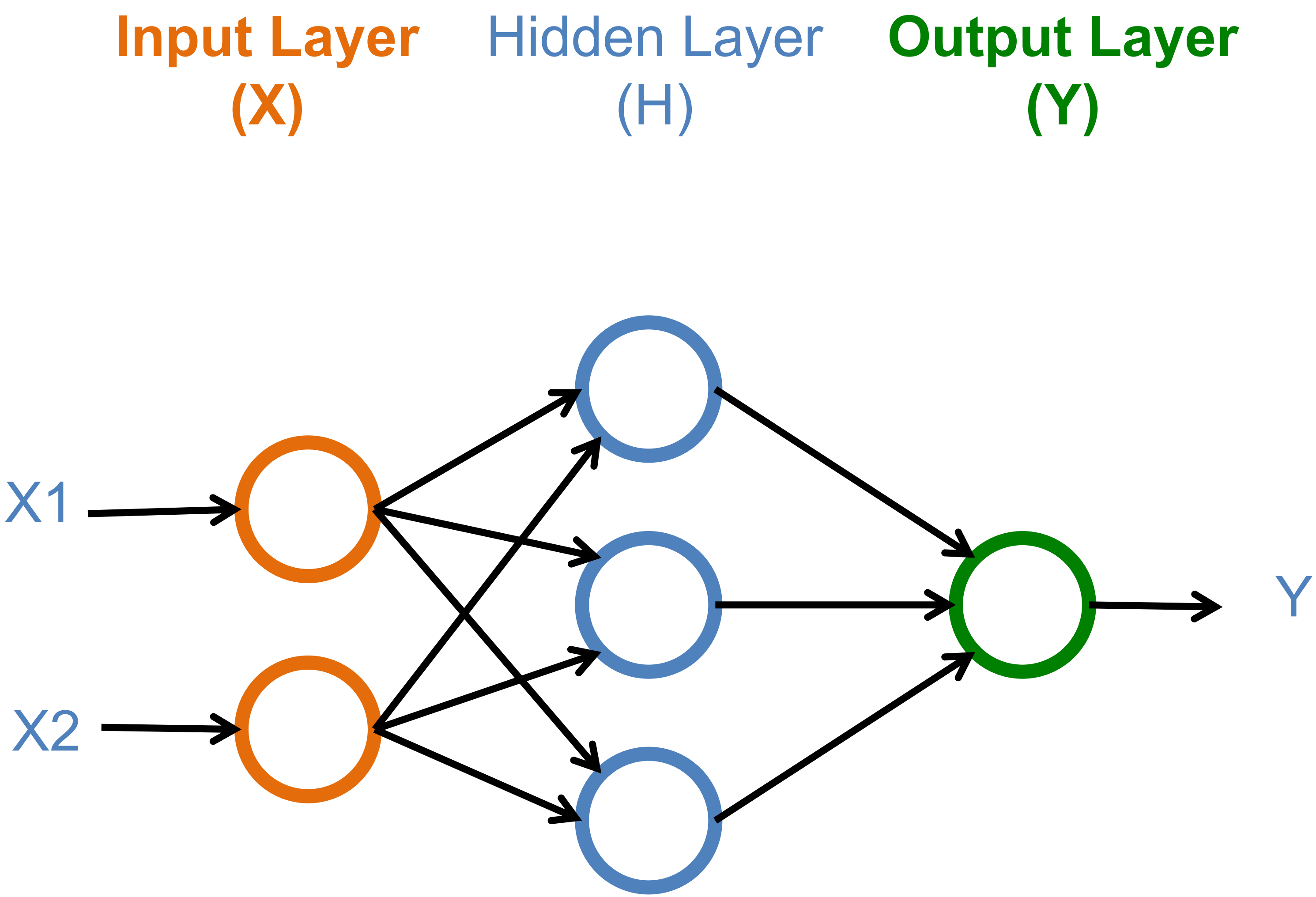
Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Neural Networks



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

Neural Networks



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

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

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

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

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

$$Y = W X + b$$

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

Output input

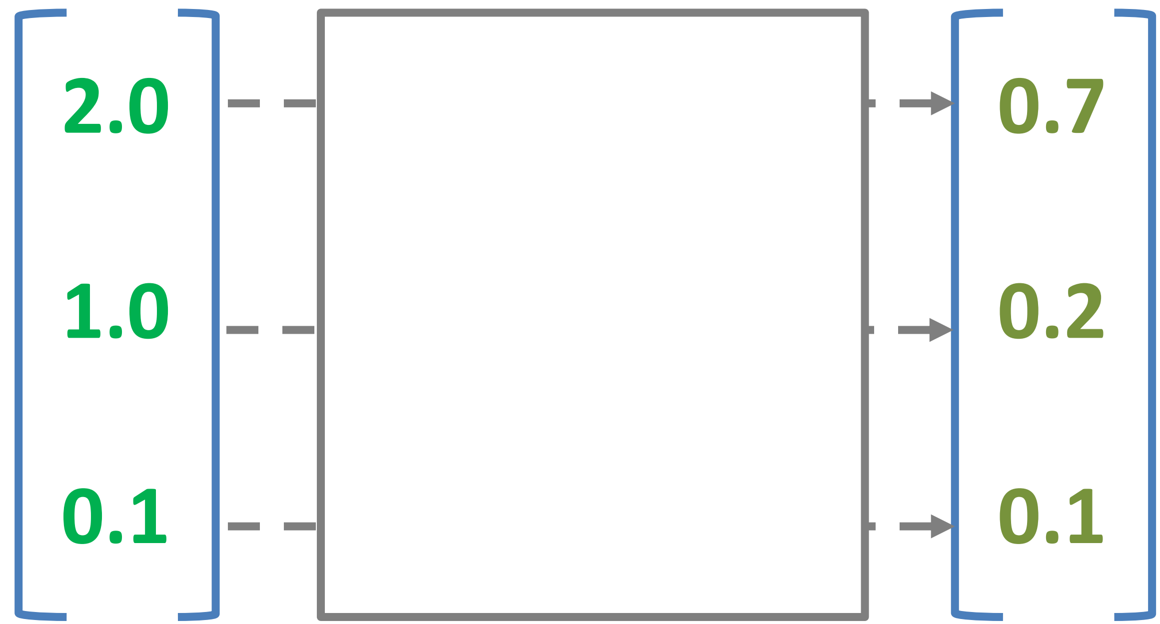
$$Y = W X + b$$

Weights bias

Trained

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

$$WX + b = Y$$

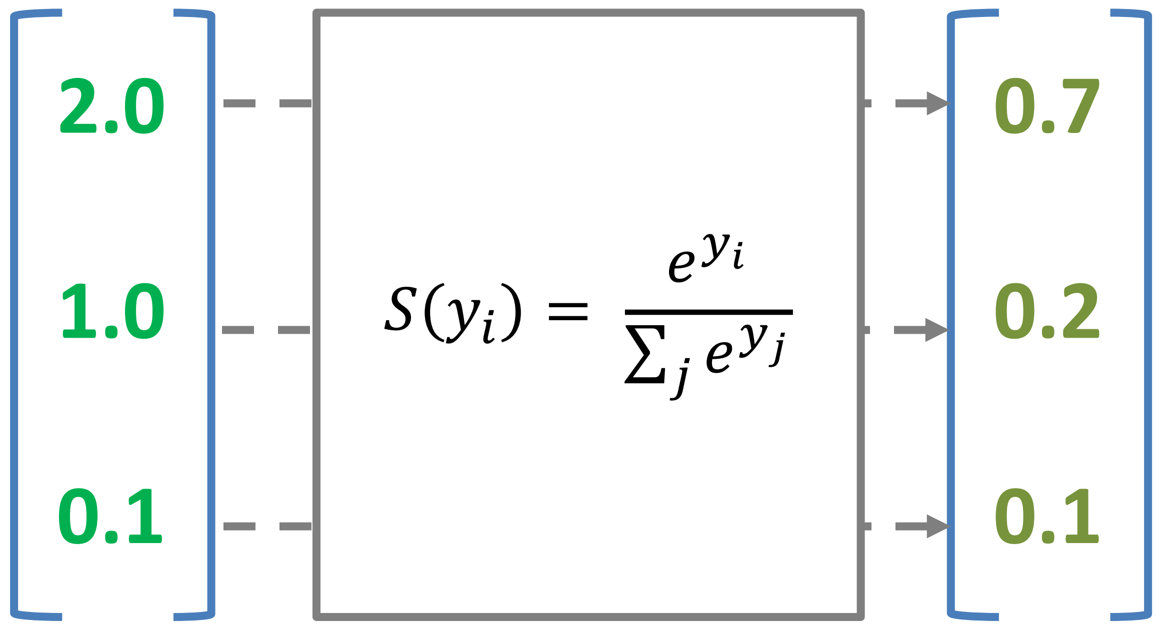


Scores → Probabilities

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

SoftMAX

W X + b = Y



Logits

Scores → Probabilities

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

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

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

W X + b = Y

$$\begin{bmatrix} 2.0 \\ 1.0 \\ 0.1 \end{bmatrix} \rightarrow \boxed{S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}}} \rightarrow \begin{bmatrix} 0.7 \\ 0.2 \\ 0.1 \end{bmatrix}$$

Logits

Scores → Probabilities

Training a Network

=

Minimize the Cost Function

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

Training a Network

=

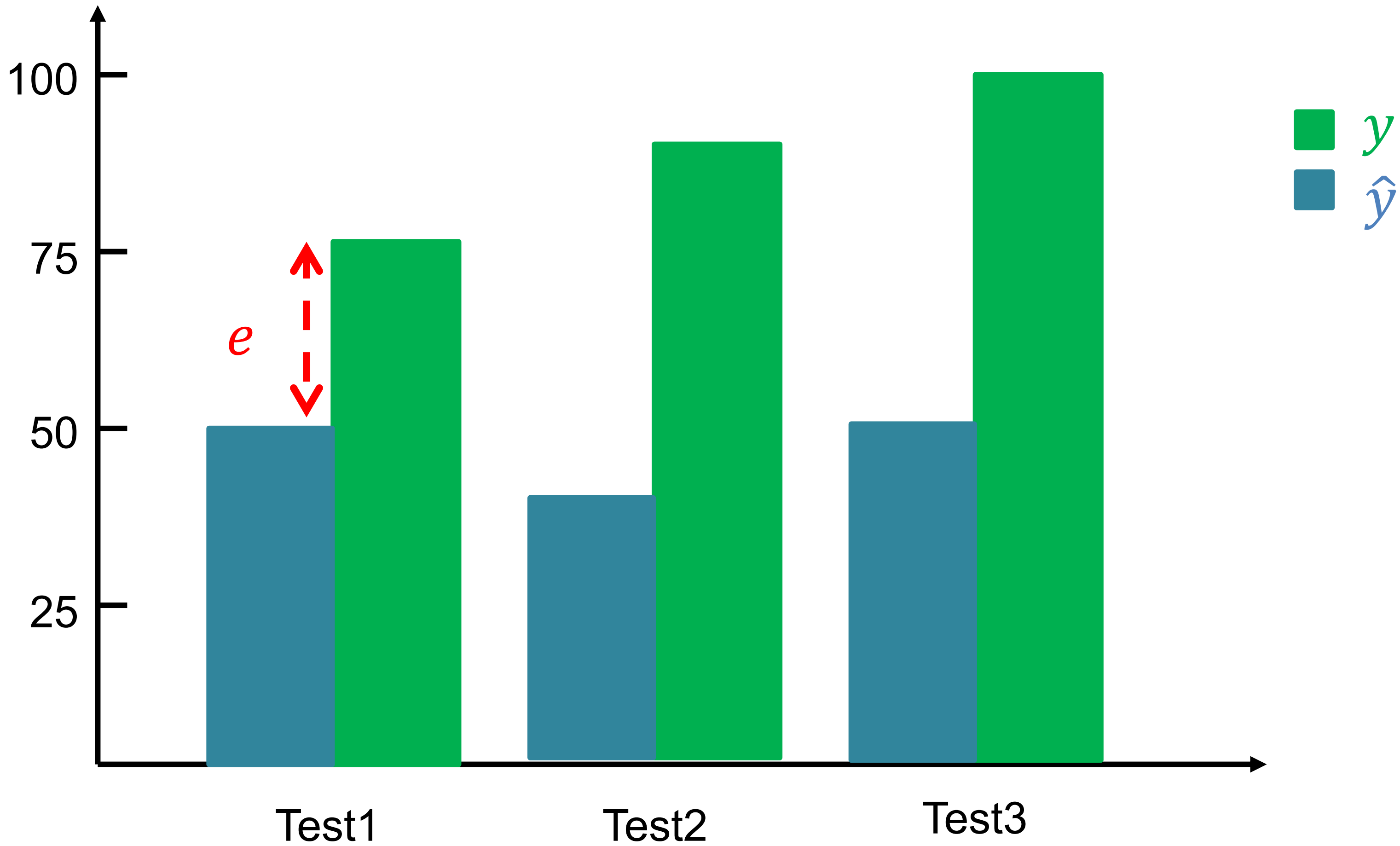
Minimize the Cost Function

Minimize the Loss Function

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

Error = Predict Y - Actual Y

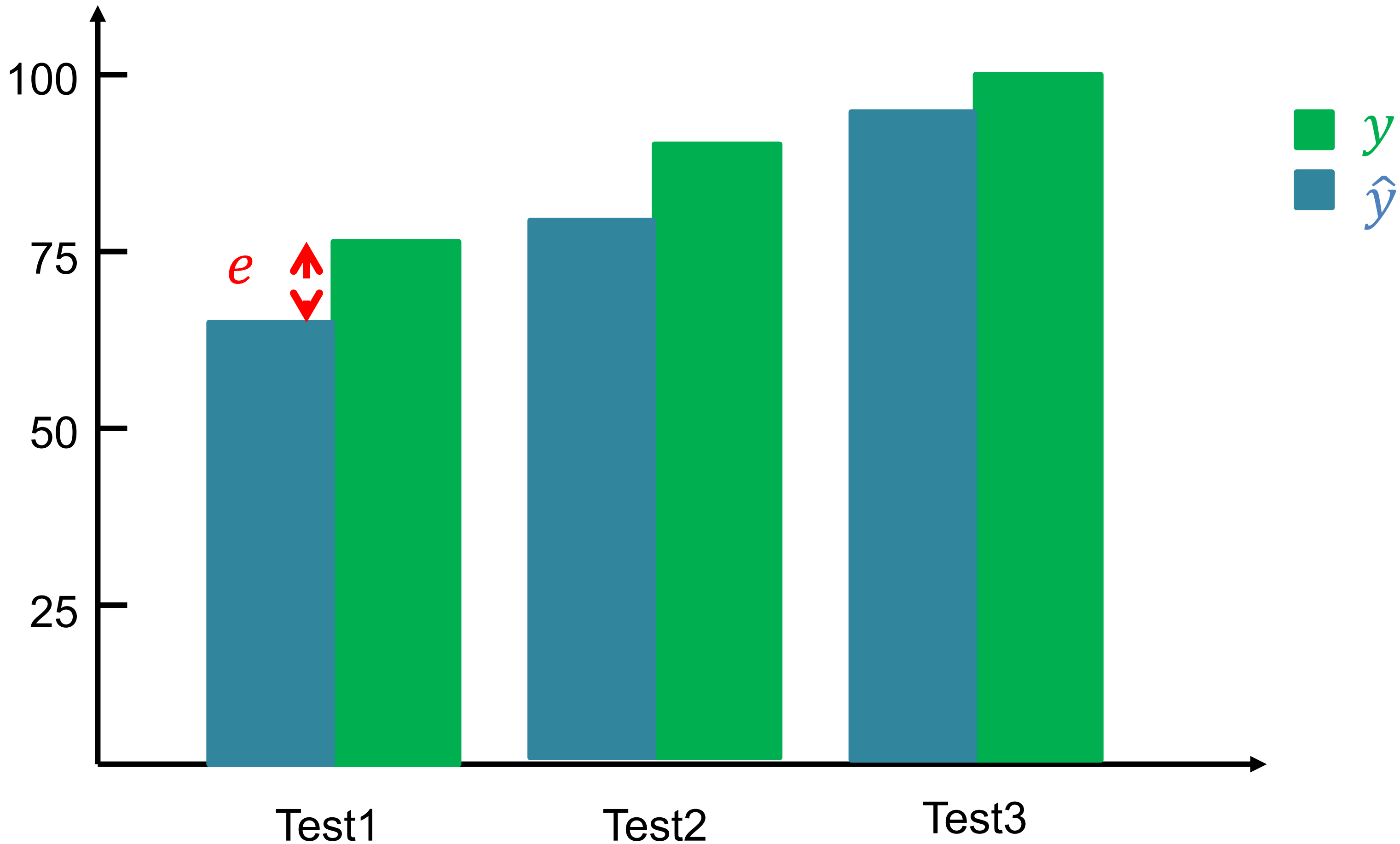
Error : Cost : Loss



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

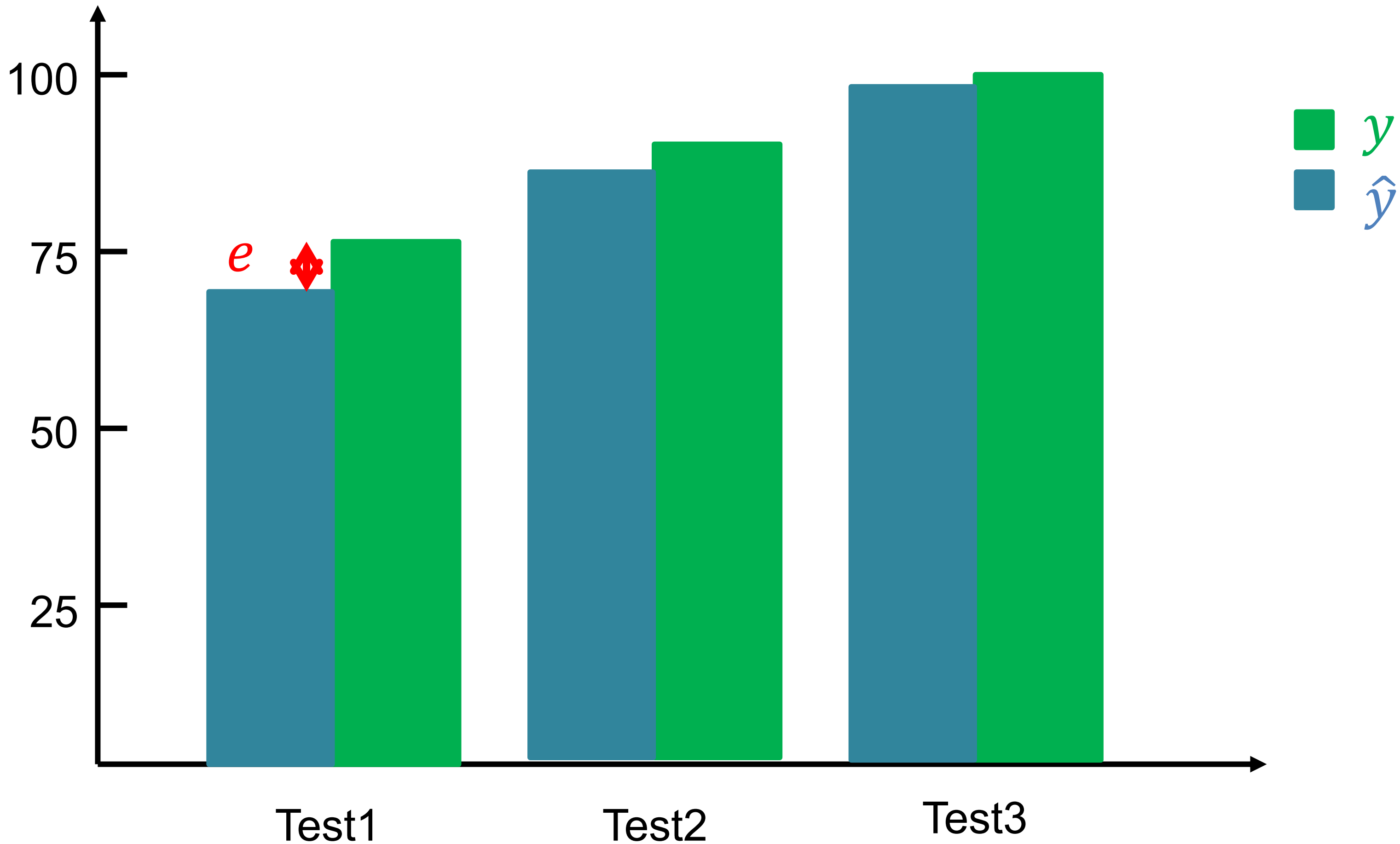
Error = Predict Y - Actual Y

Error : Cost : Loss



Error = Predict Y - Actual Y

Error : Cost : Loss

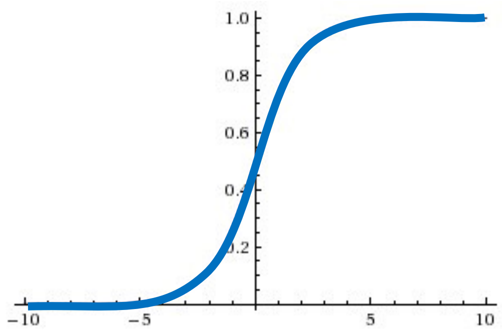


Source: https://www.youtube.com/watch?v=bxe2T-V8XRs&index=1&list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU

Activation Functions

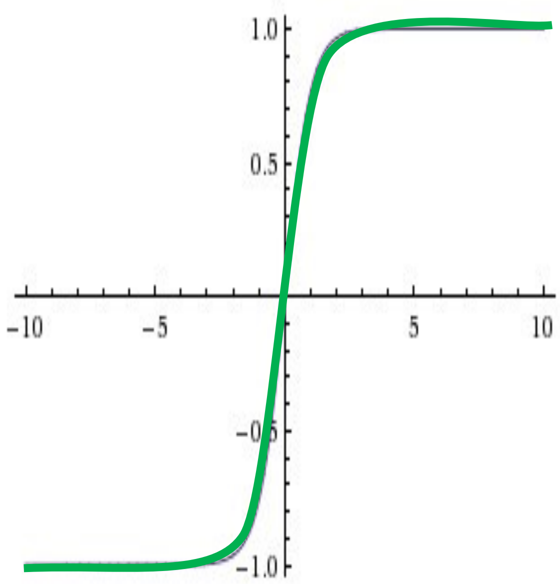
Activation Functions

Sigmoid



[0, 1]

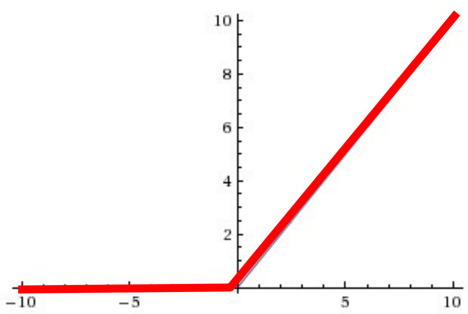
TanH



[-1, 1]

ReLU

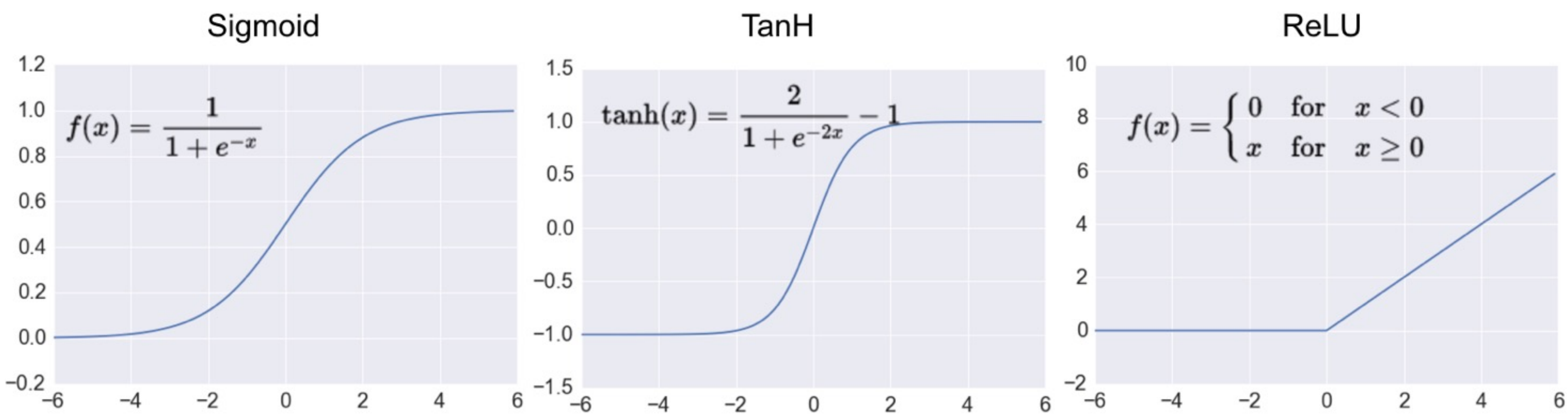
(Rectified Linear Unit)



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

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

Activation Functions



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

Loss Function

Binary Classification: 2 Class

Activation Function:
Sigmoid

Loss Function:
Binary Cross-Entropy

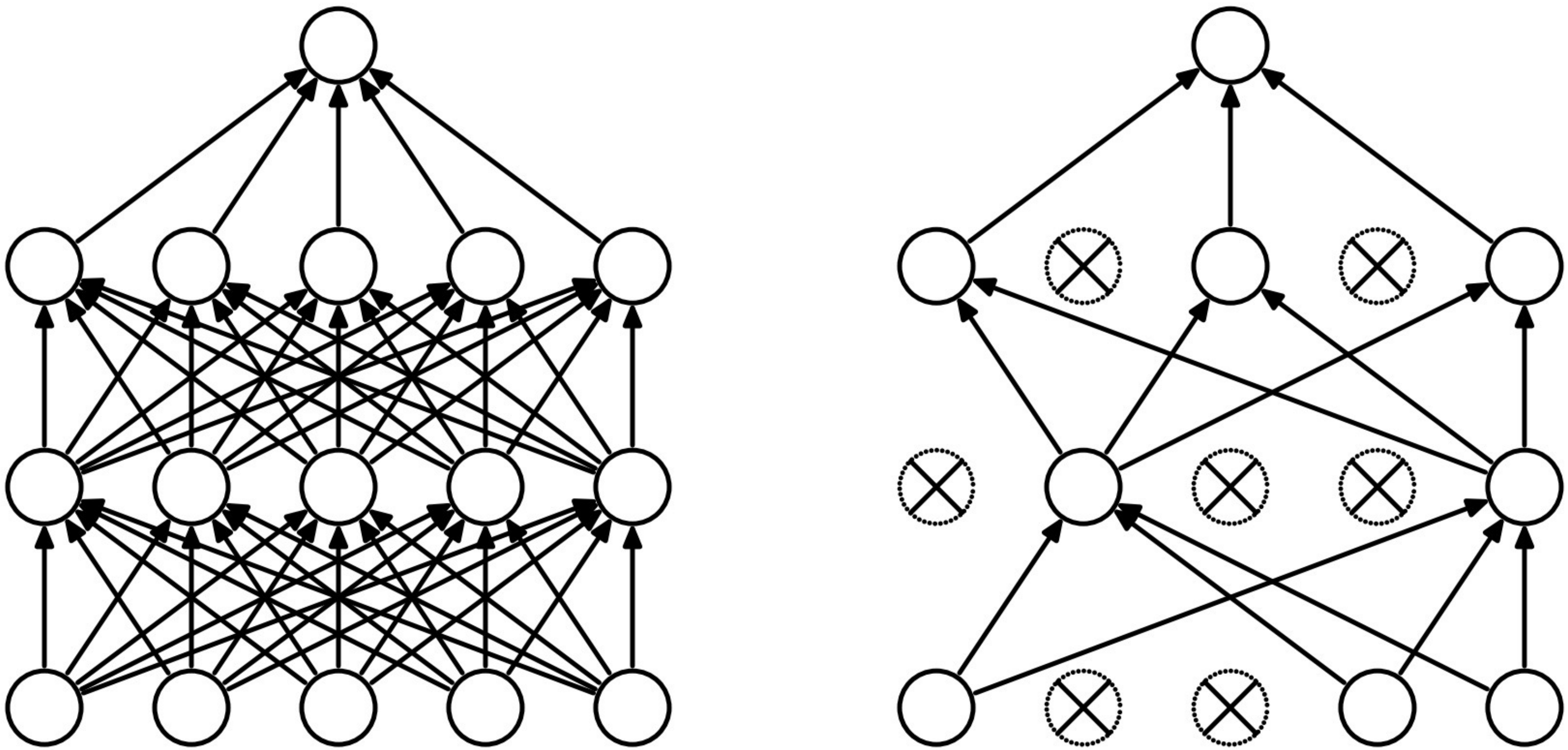
Multiple Classification: 10 Class

Activation Function:
SoftMAX

Loss Function:
Categorical Cross-Entropy

Dropout

Dropout: a simple way to prevent neural networks from overfitting



(a) Standard Neural Net (b) After applying dropout.

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

Learning Algorithm

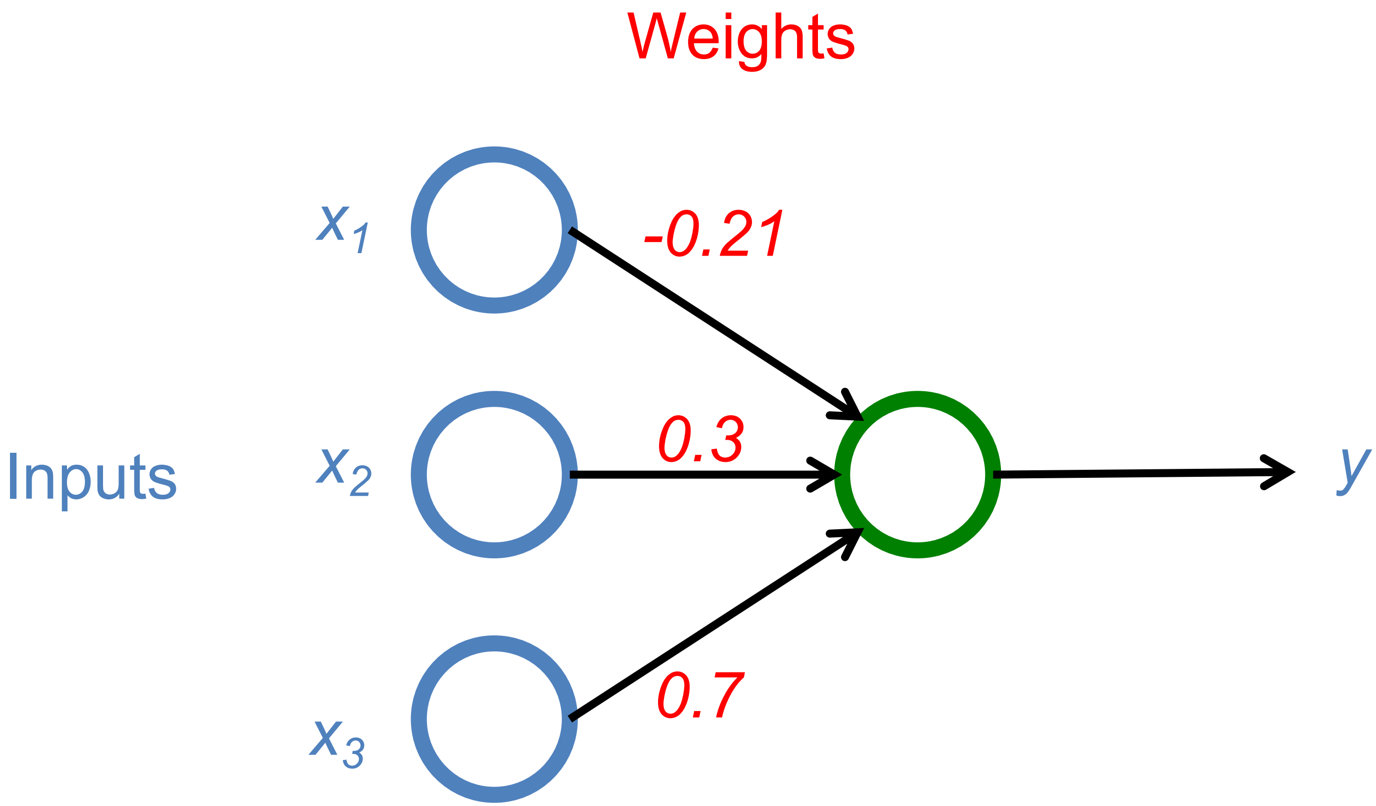
While not done:

Pick a random training example "(input, label)"

Run neural network on "input"

Adjust weights on edges to make output closer to "label"

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



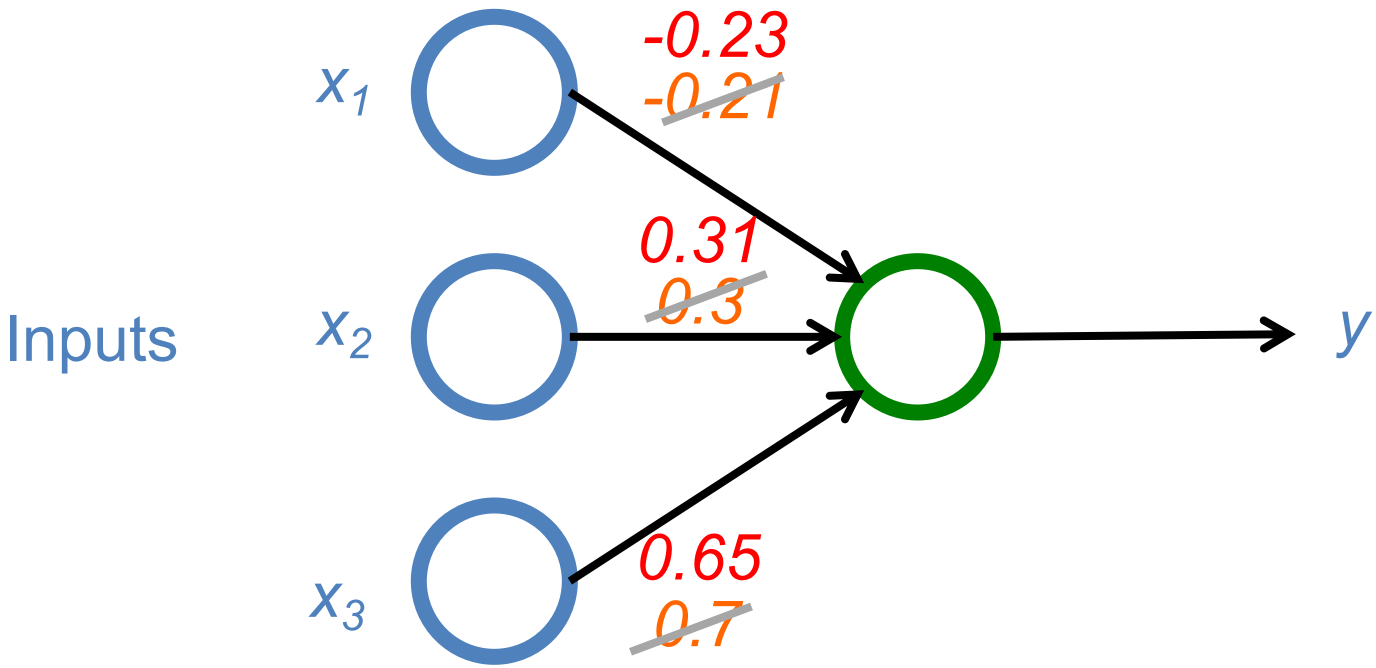
Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

Next time:

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

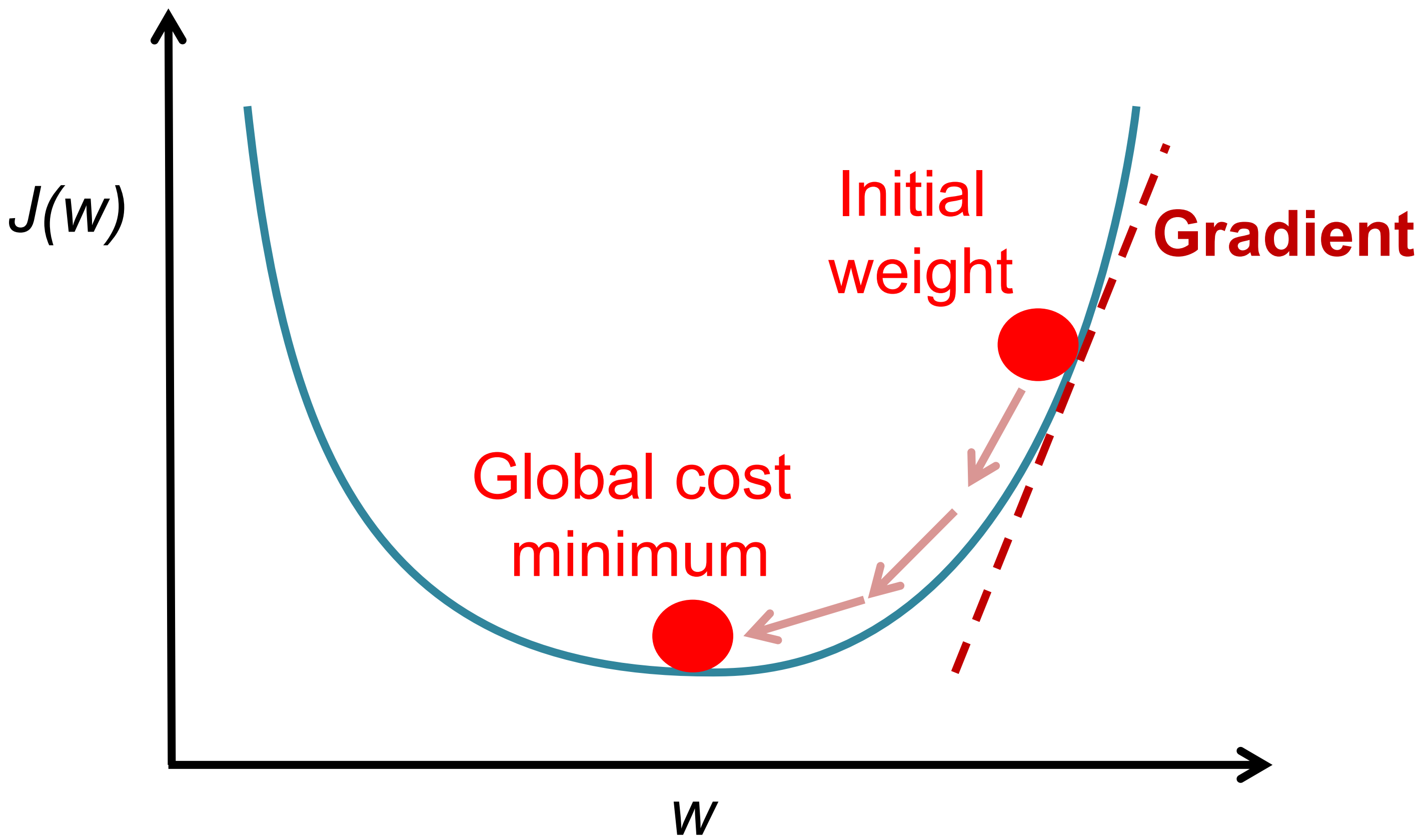
$$\cancel{y = max\ (\ 0,\ -0.21\ *\ x_1 + 0.3\ *\ x_2 + 0.7\ *\ x_3\ )}$$

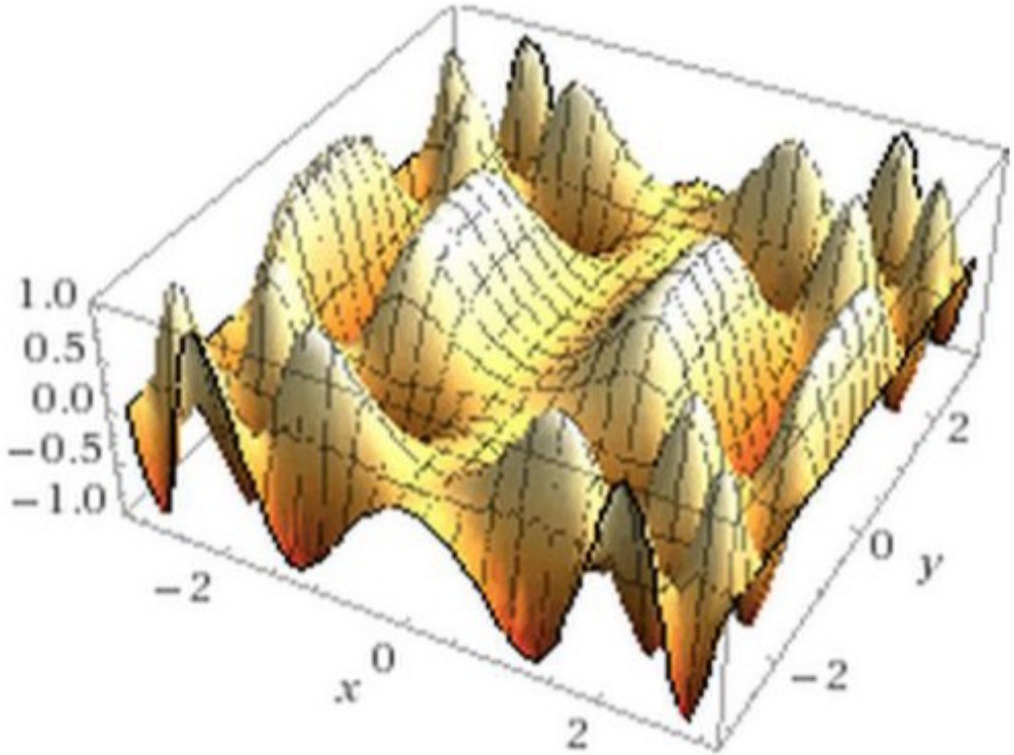
Weights



Optimizer:

Stochastic Gradient Descent (SGD)

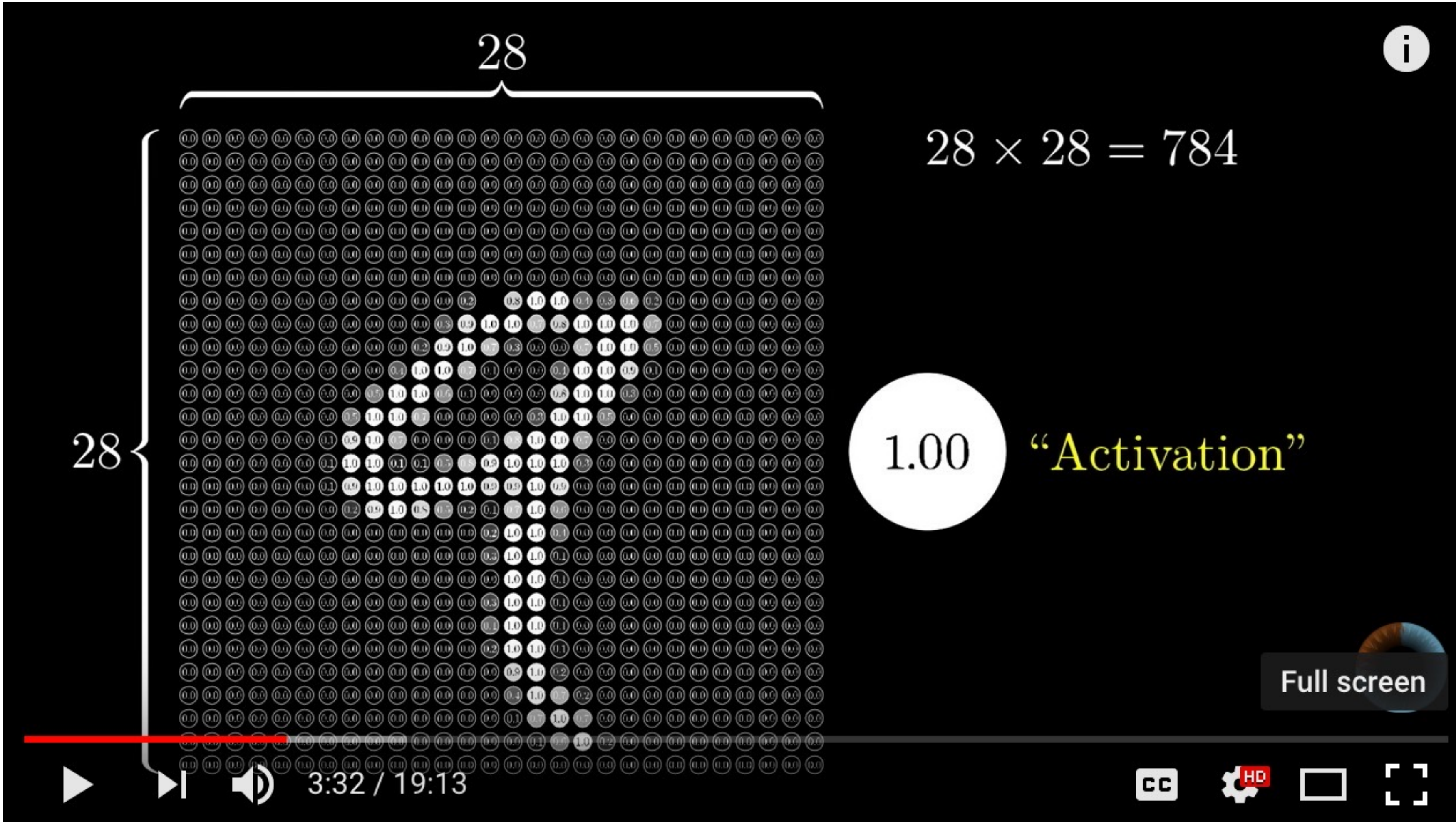




This shows a function of 2 variables: real neural nets are functions of hundreds of millions of variables!

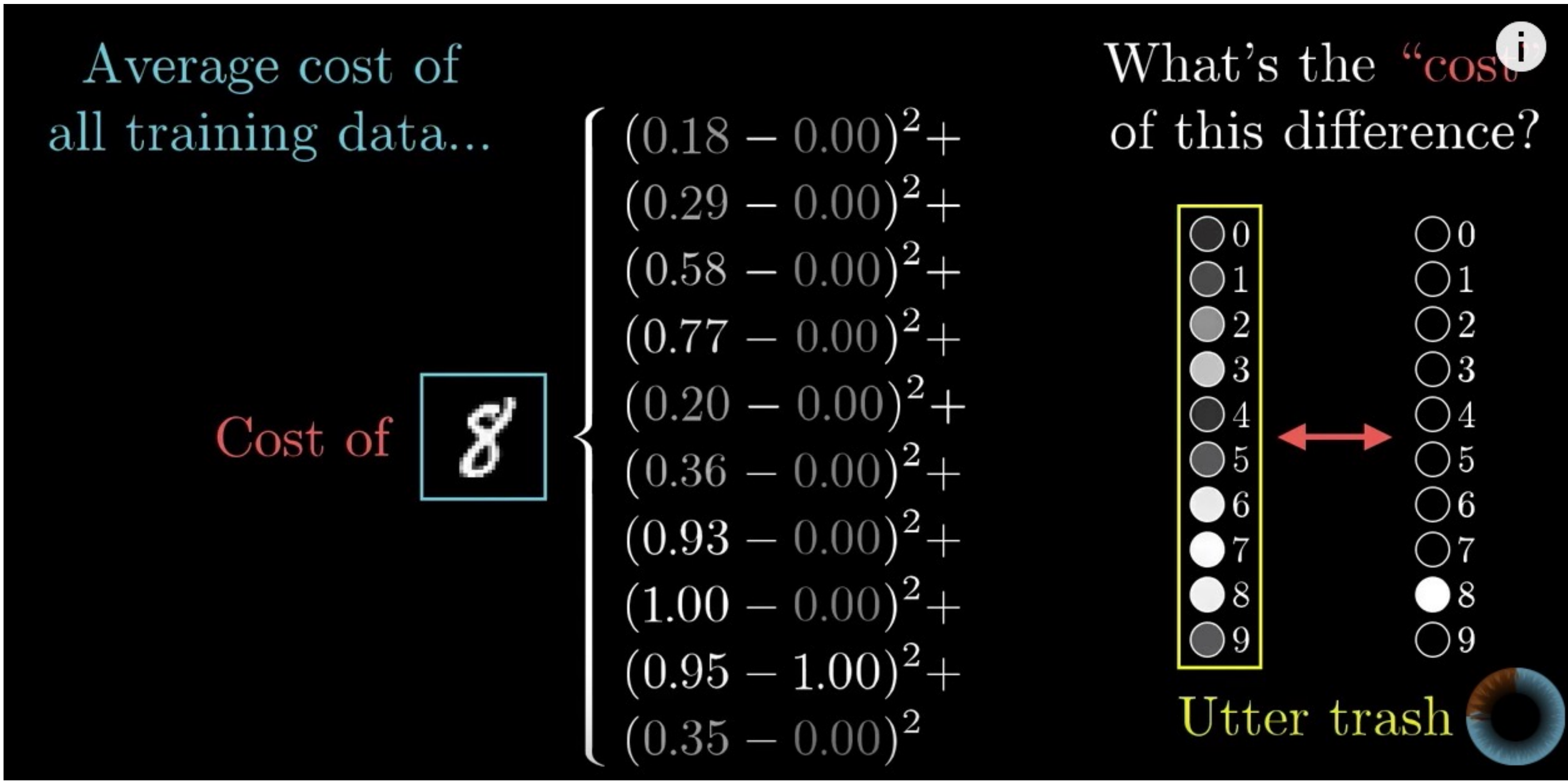
Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

Neural Network and Deep Learning



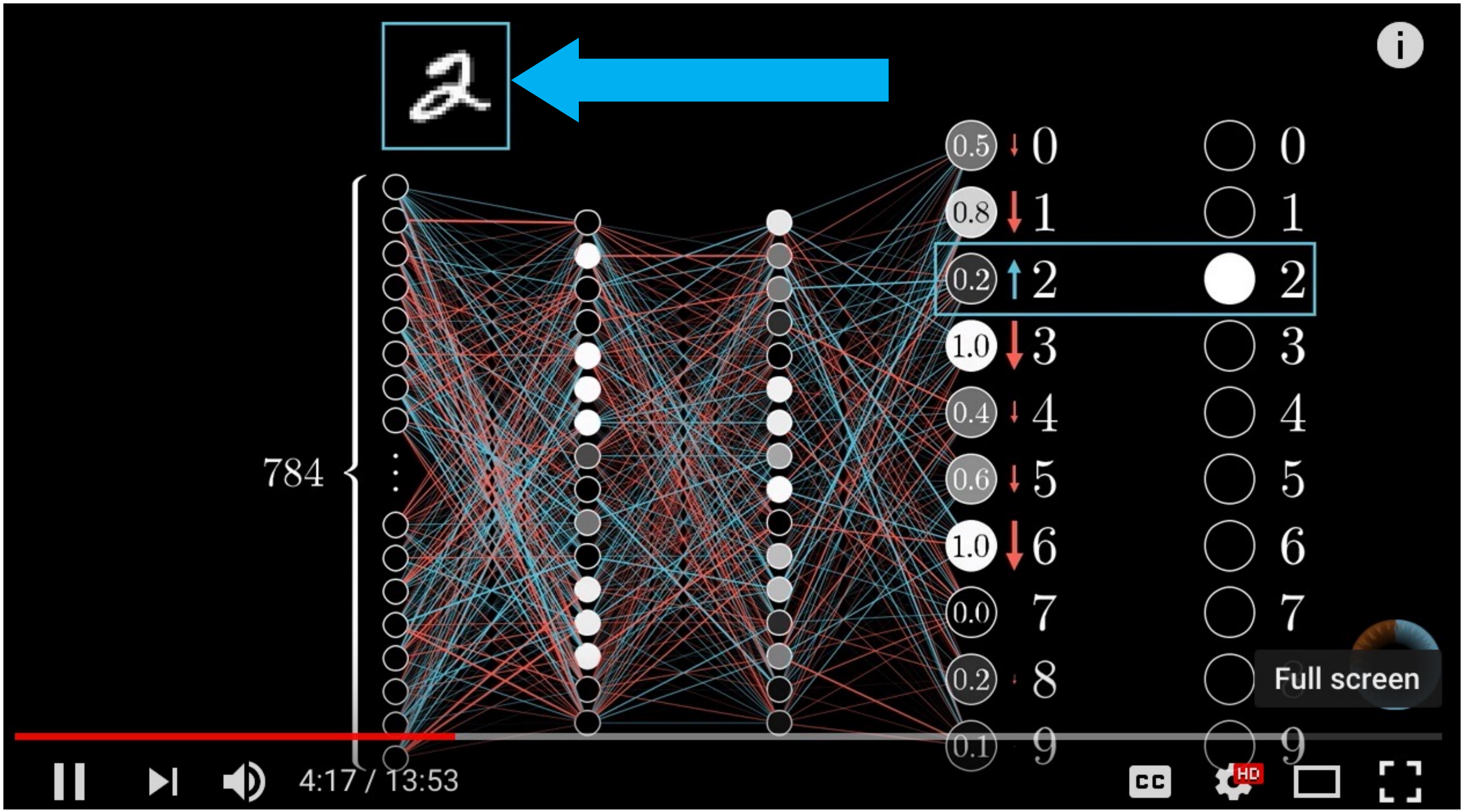
Source: 3Blue1Brown (2017), But what *is* a Neural Network? | Chapter 1, deep learning, https://www.youtube.com/watch?v=aircAruvnKk

Gradient Descent
how neural networks learn



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

Backpropagation



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

Learning Algorithm

While not done:

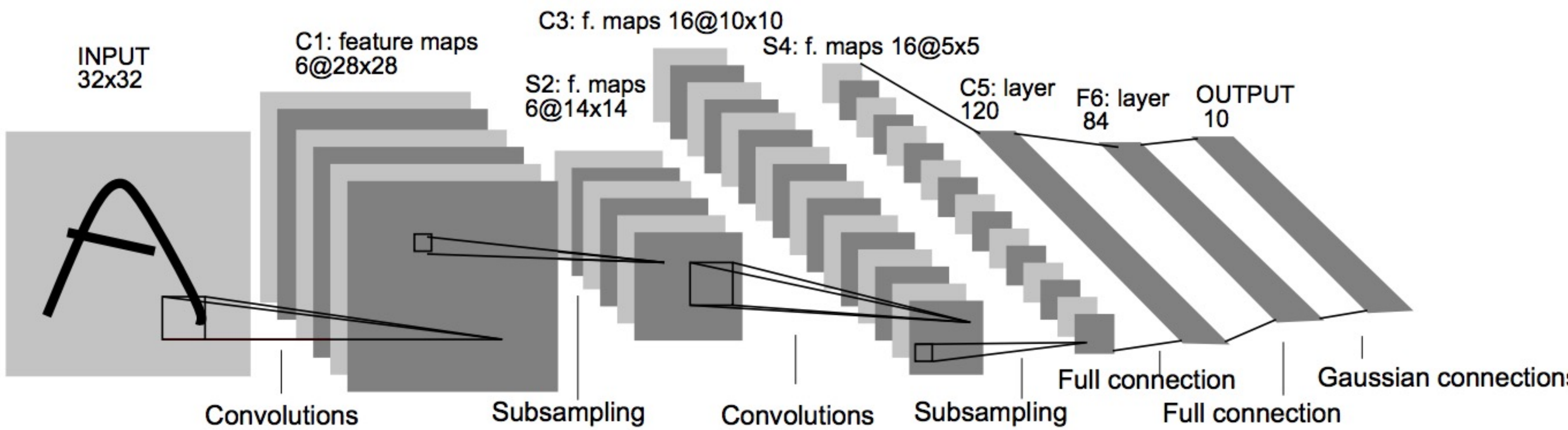
Pick a random training example "(input, label)"

Run neural network on "input"

Adjust weights on edges to make output closer to "label"

Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN)



Architecture of LeNet-5 (7 Layers) (LeCun et al., 1998)

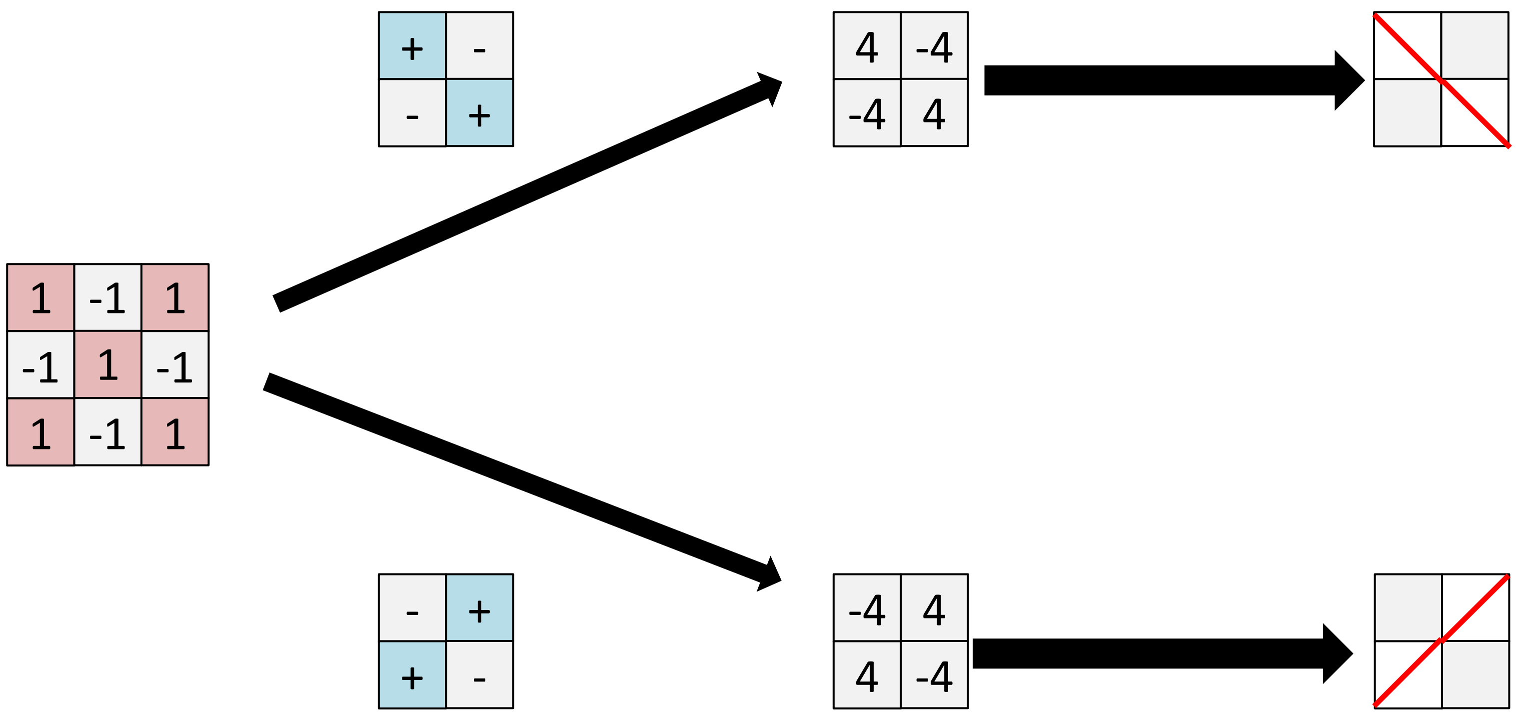
Source: http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf

Source: LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86, no. 11 (1998): 2278-2324.

Convolutional Neural Networks (CNN)

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

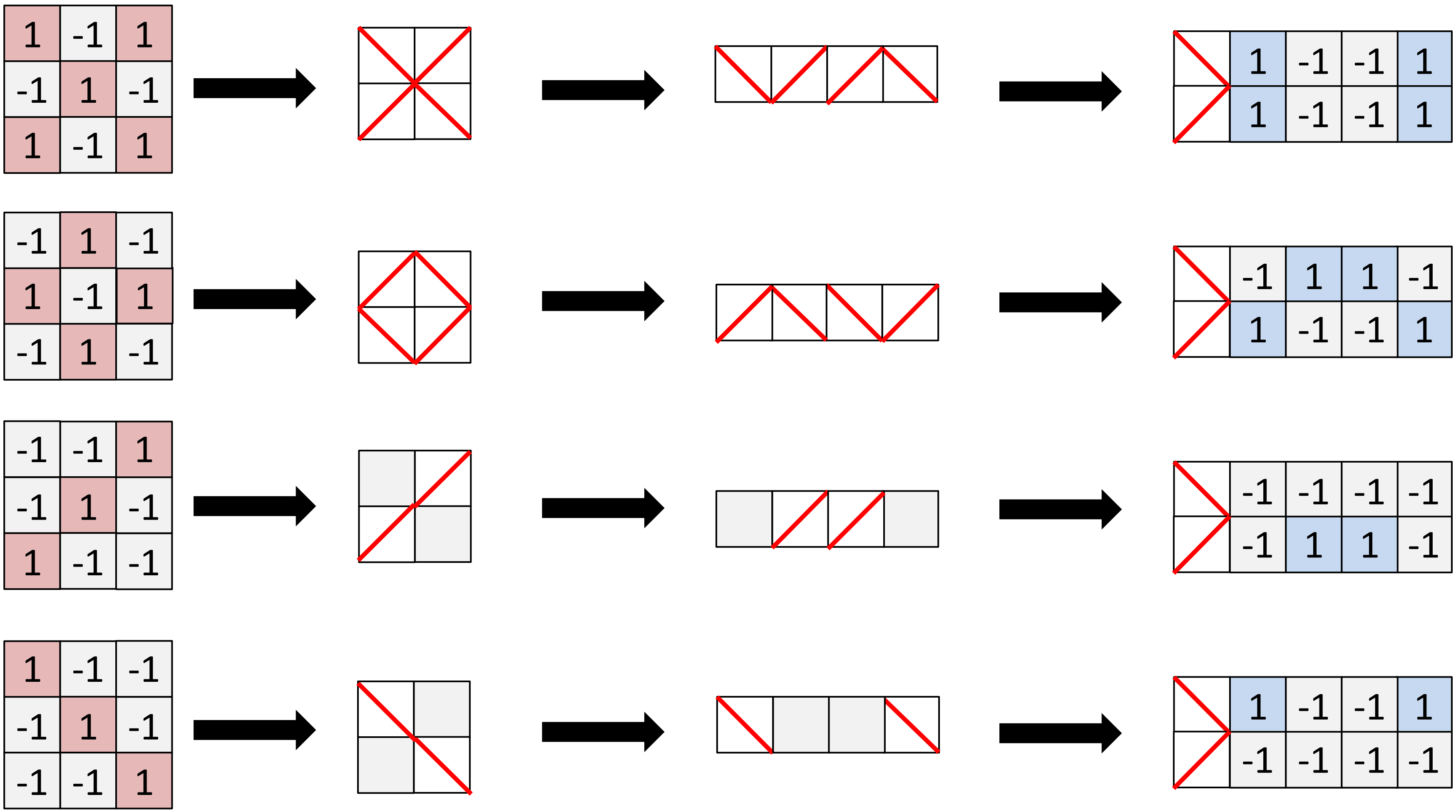
A friendly introduction to Convolutional Neural Networks and Image Recognition



Convolution Layer   Pooling Layer

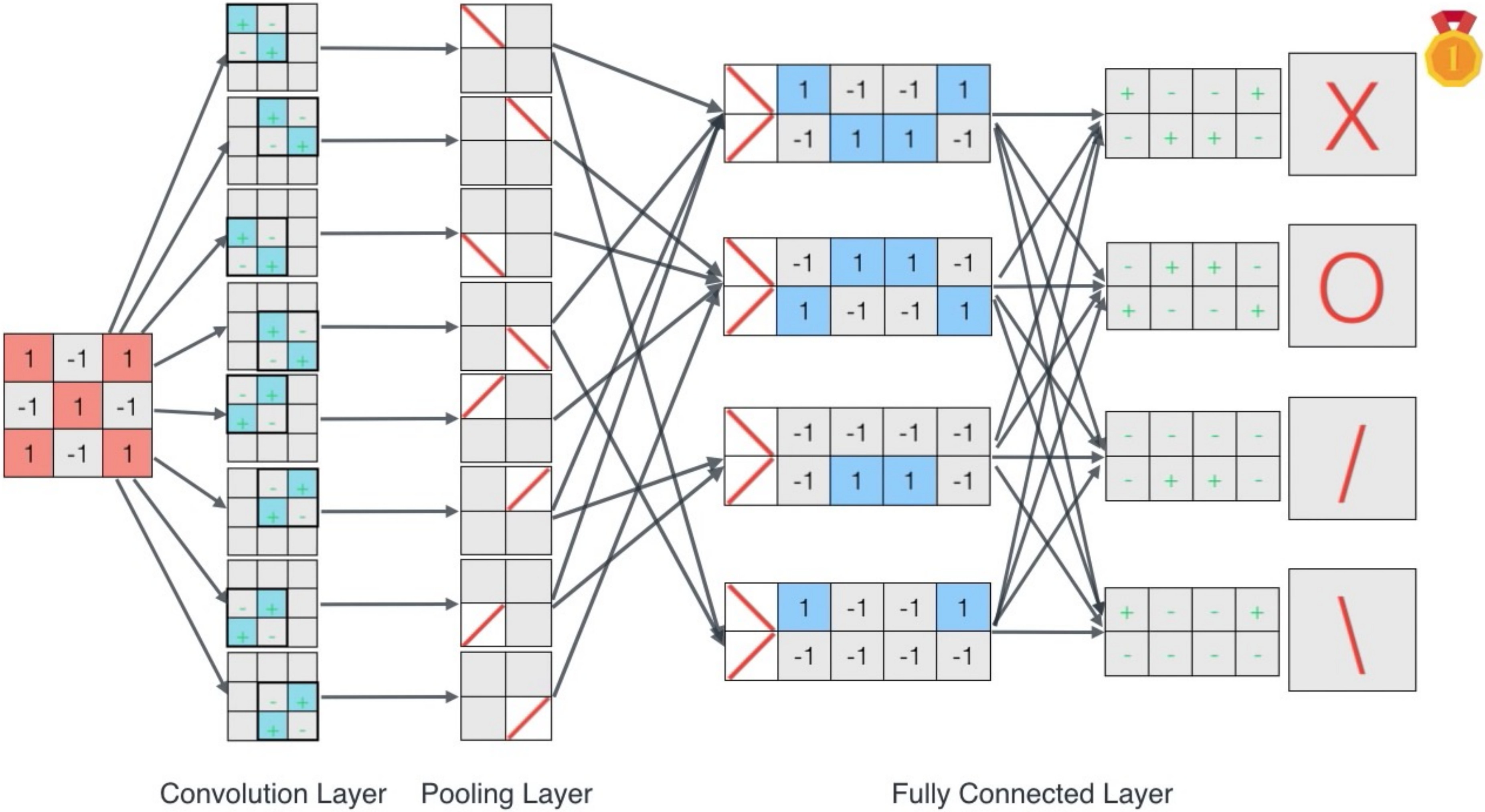
Source: Luis Serrano (2017), A friendly introduction to Convolutional Neural Networks and Image Recognition, https://www.youtube.com/watch?v=2-Ol7ZB0MmU

A friendly introduction to Convolutional Neural Networks and Image Recognition



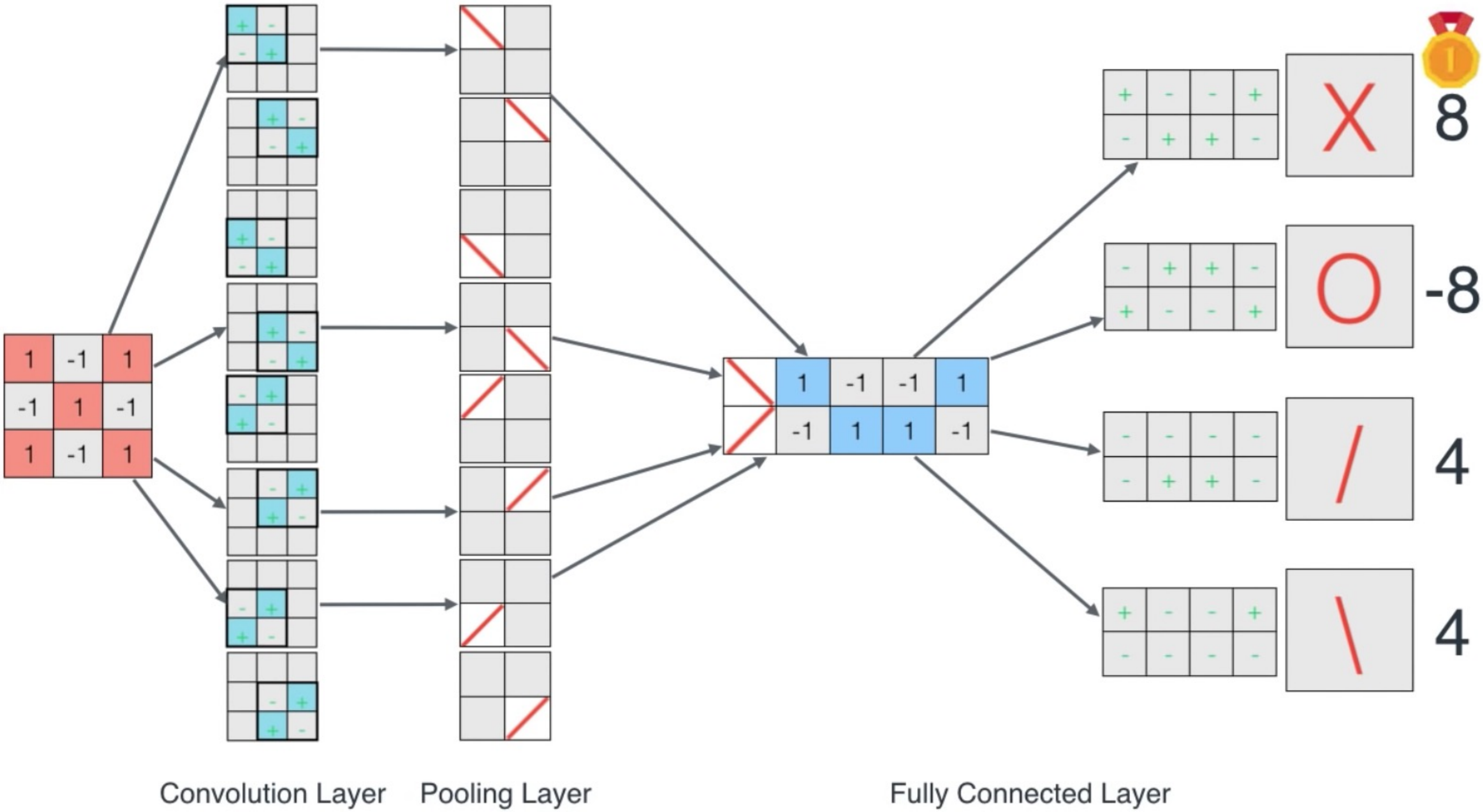
Source: Luis Serrano (2017), A friendly introduction to Convolutional Neural Networks and Image Recognition, https://www.youtube.com/watch?v=2-Ol7ZB0MmU

A friendly introduction to Convolutional Neural Networks and Image Recognition



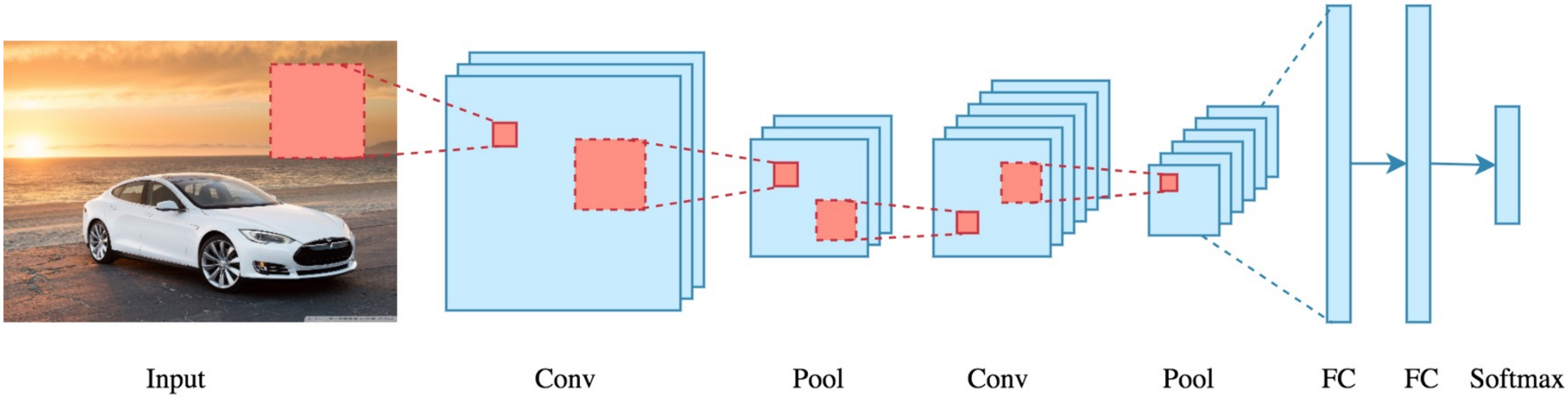
Source: Luis Serrano (2017), A friendly introduction to Convolutional Neural Networks and Image Recognition, https://www.youtube.com/watch?v=2-Ol7ZB0MmU

A friendly introduction to Convolutional Neural Networks and Image Recognition



Source: Luis Serrano (2017), A friendly introduction to Convolutional Neural Networks and Image Recognition, https://www.youtube.com/watch?v=2-Ol7ZB0MmU

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

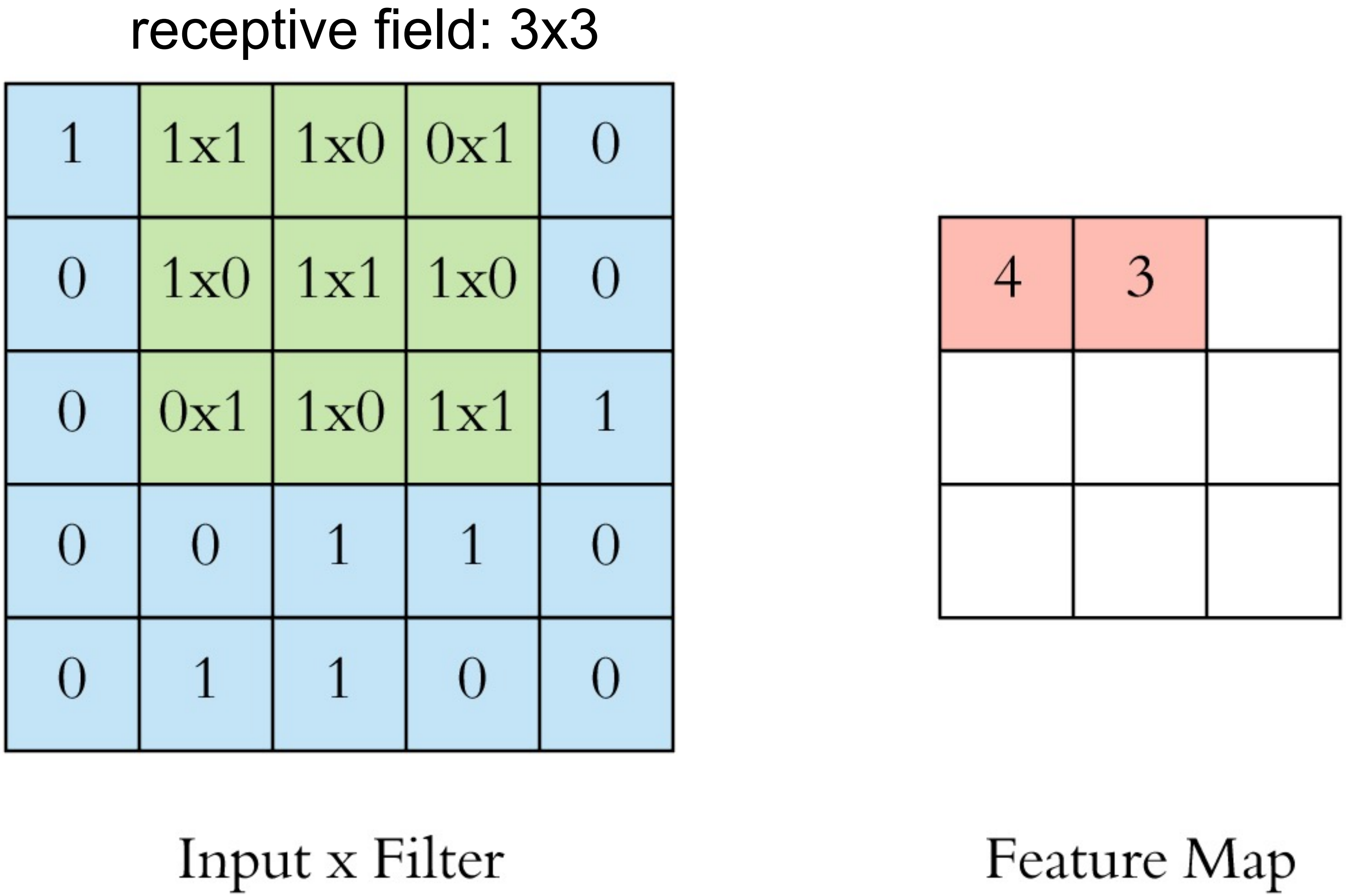
Input x Filter

4		

Feature Map

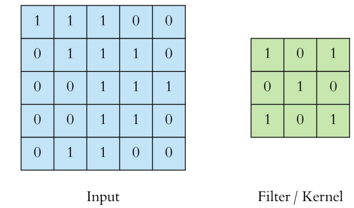
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

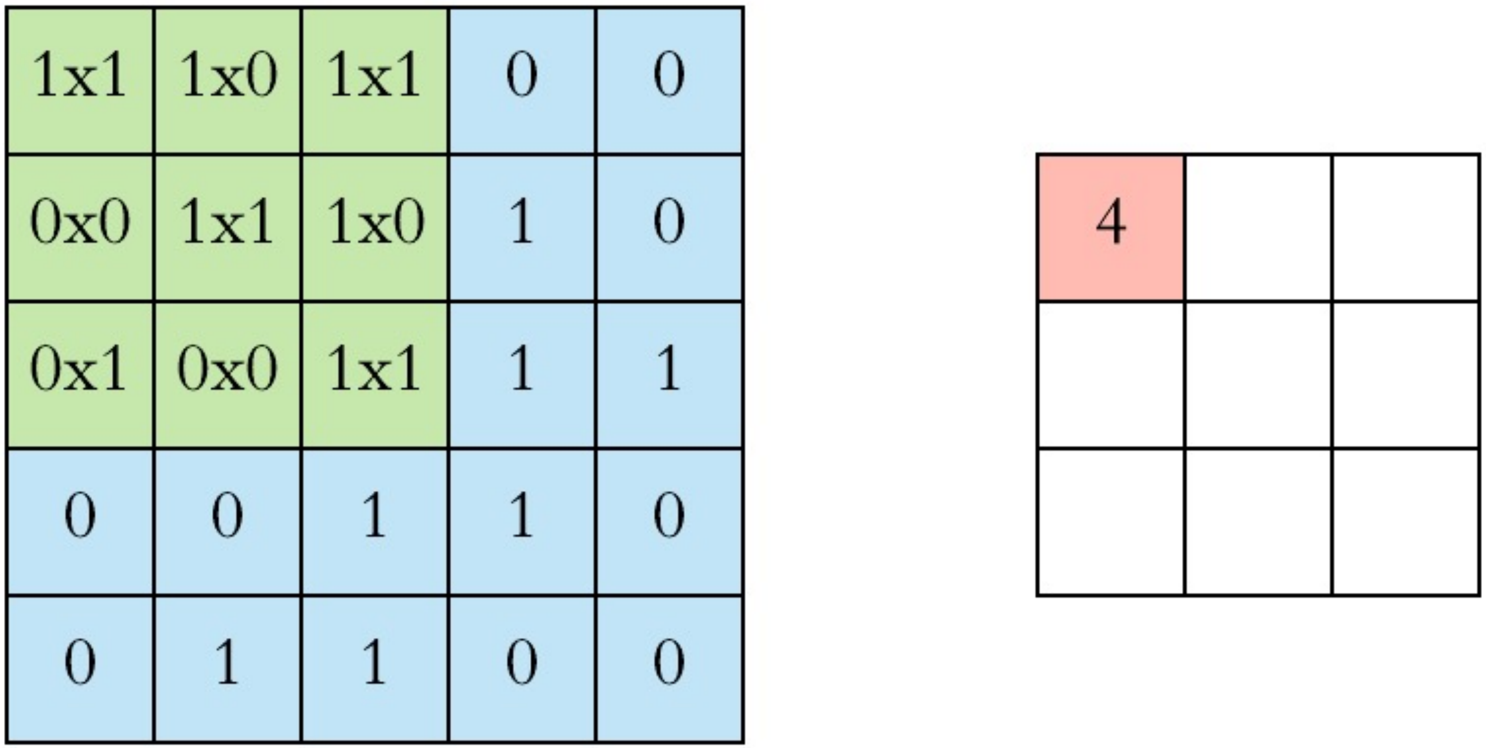
CNN Convolution Layer Input x Filter --> Feature Map



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

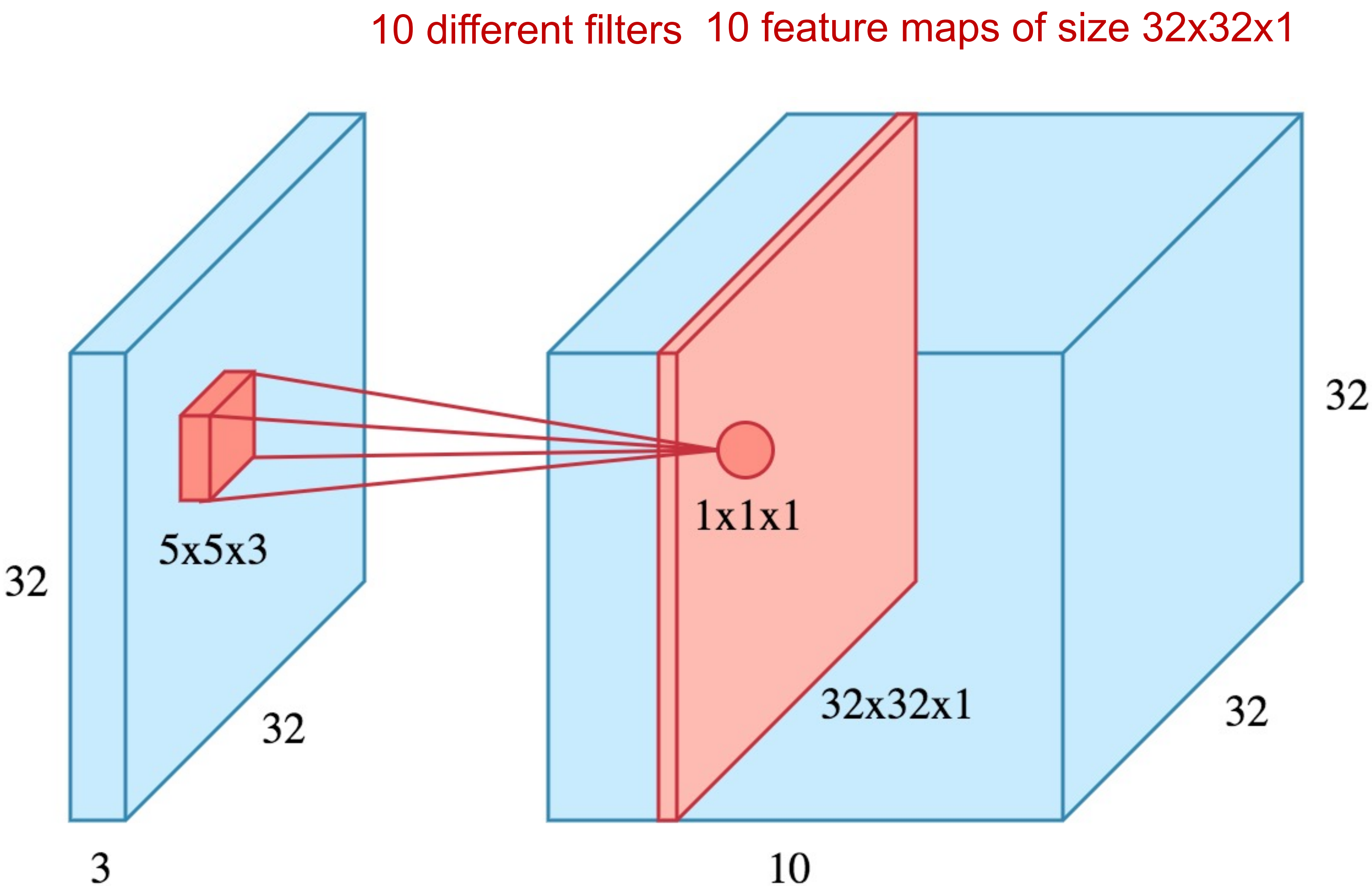




Example convolution operation shown in 2D using a 3x3 filter

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

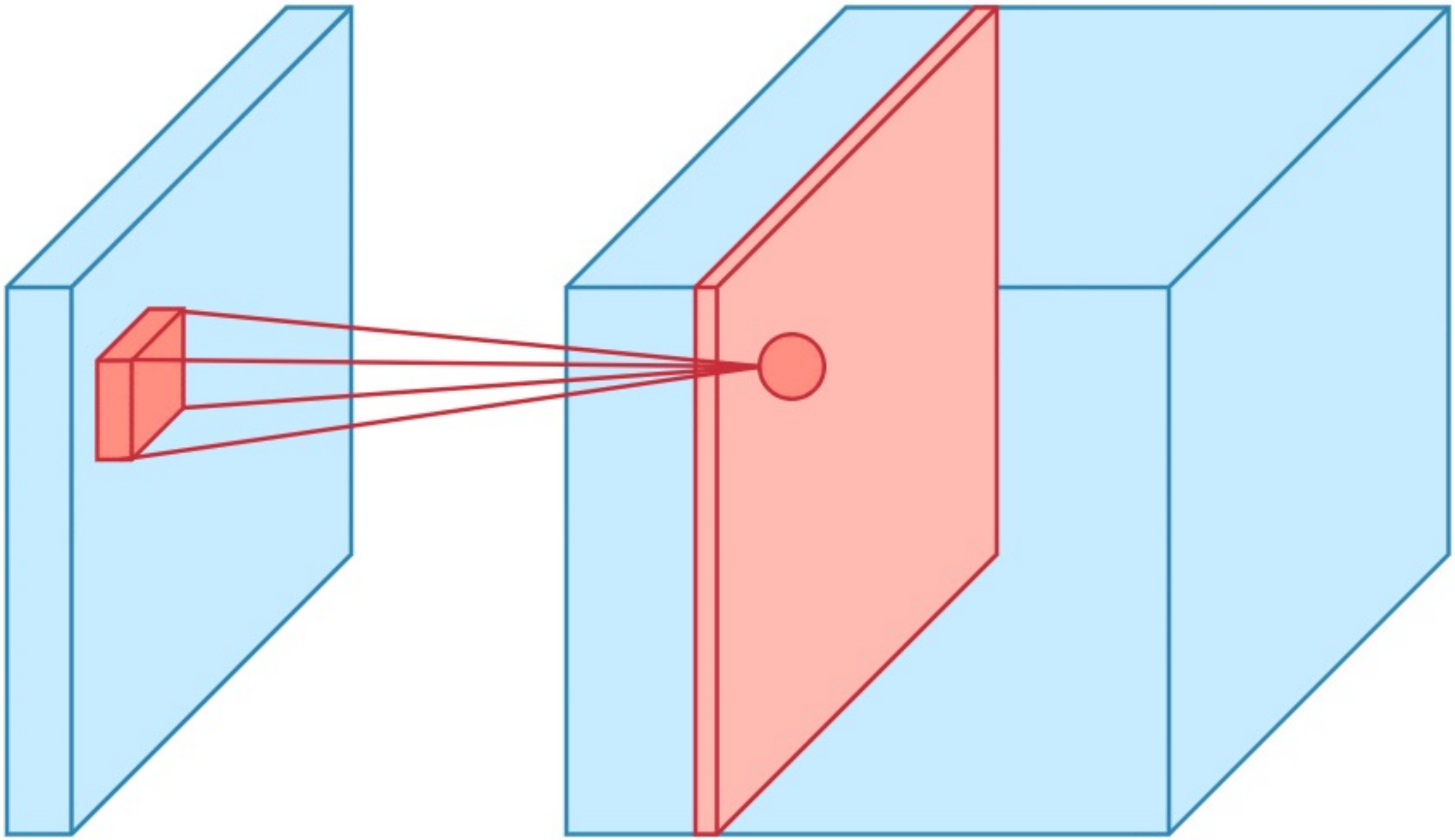
CNN Convolution Layer



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

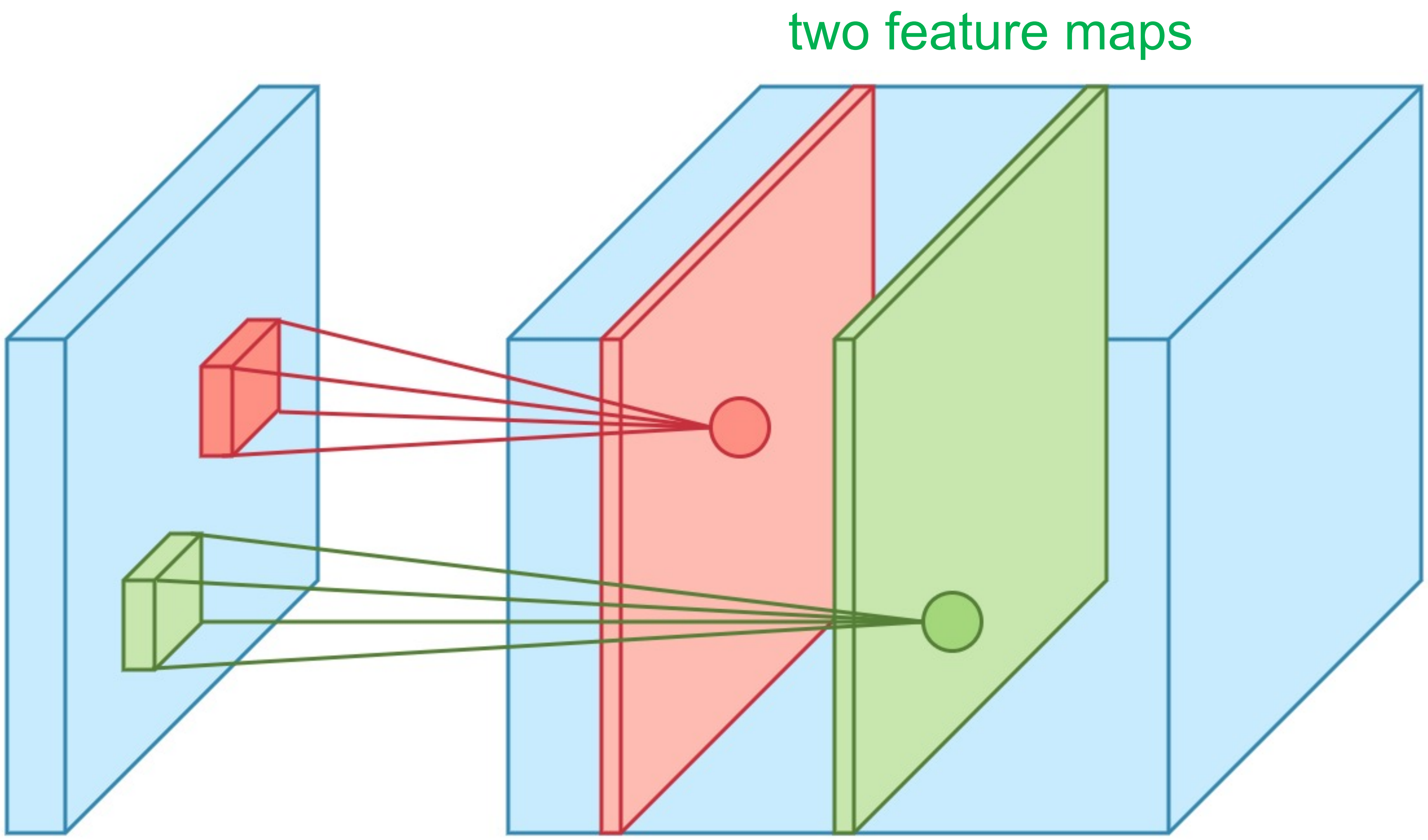
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks,
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

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

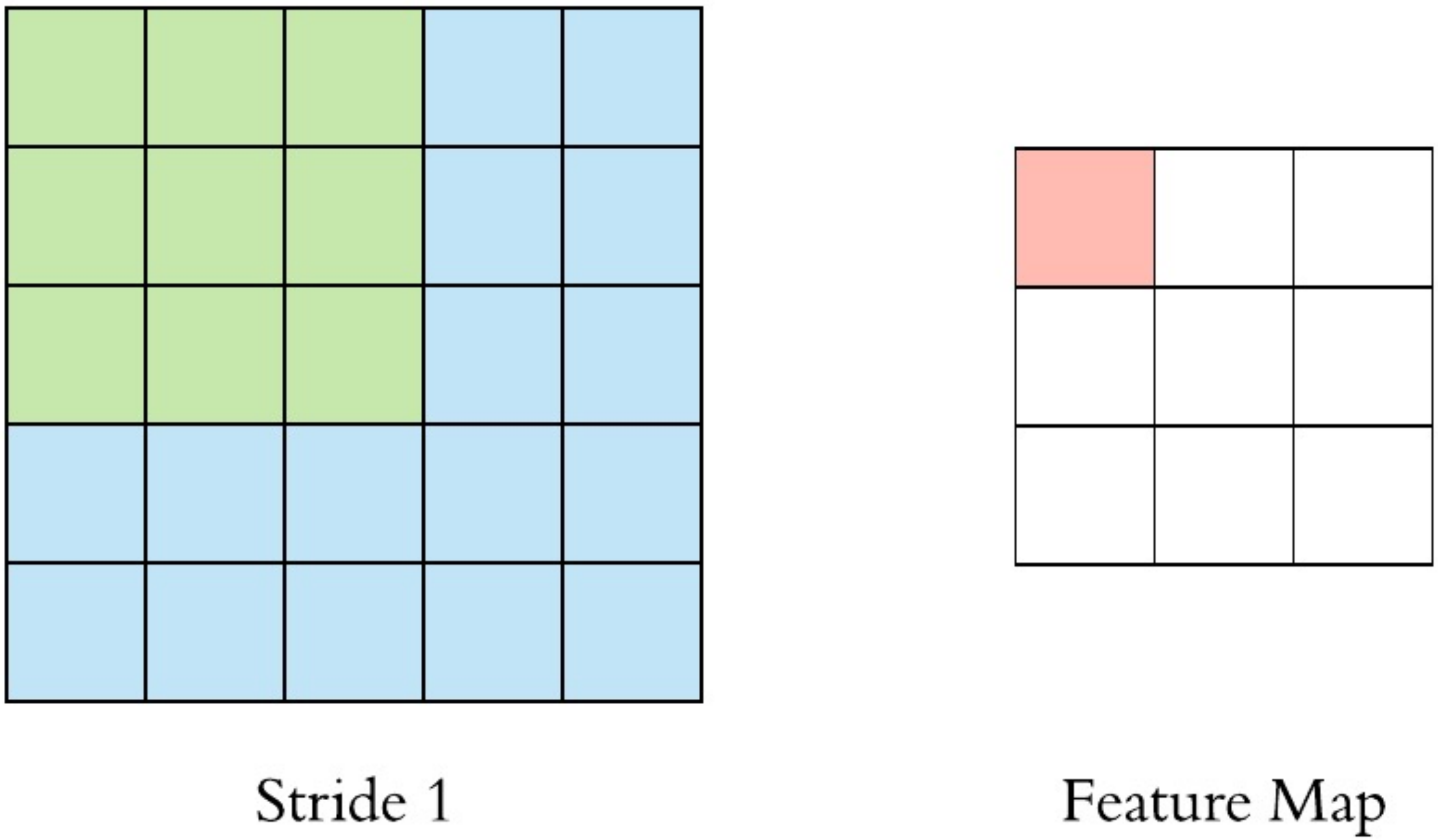
CNN Convolution Layer



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

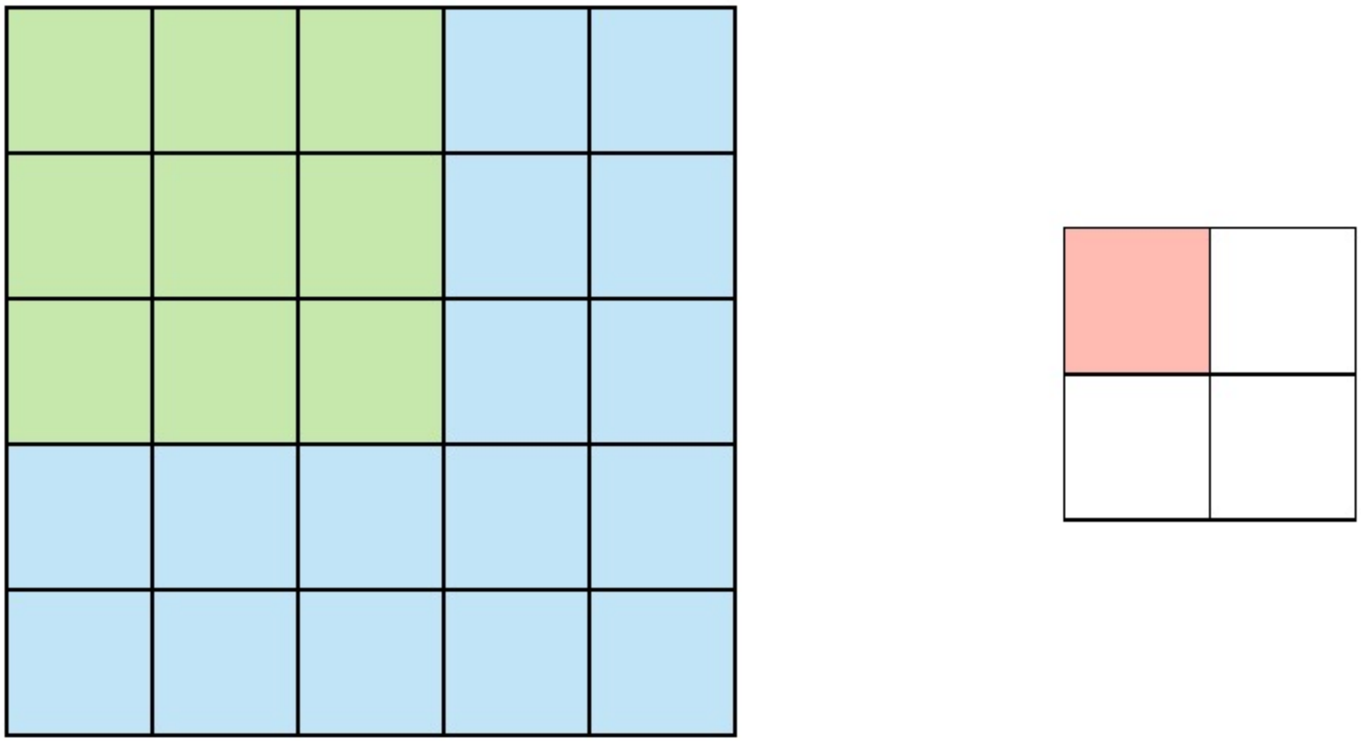
Stride specifies how much
we move the convolution filter at each step



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

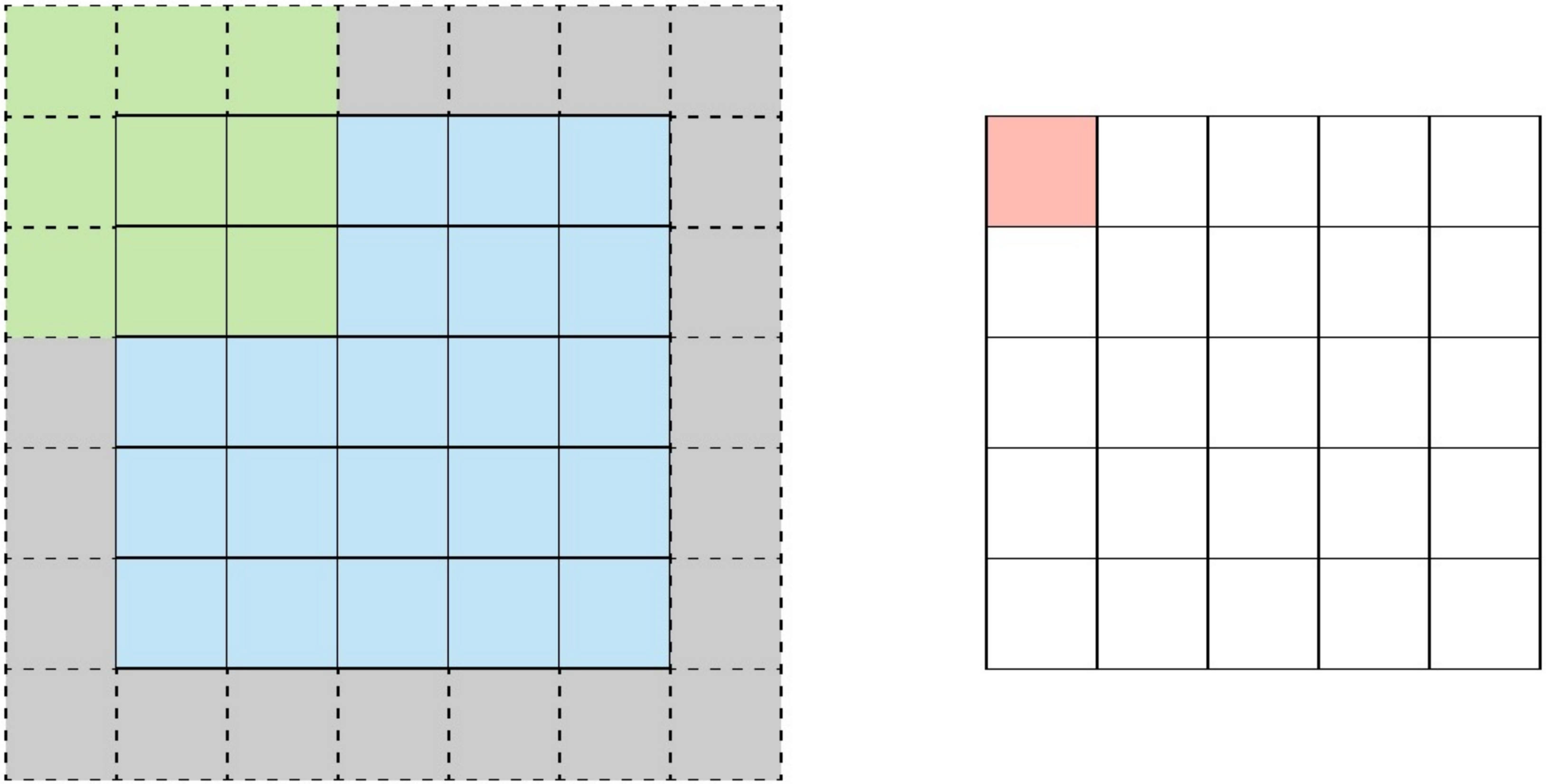
Stride specifies how much
we move the convolution filter at each step



Stride 2 Feature Map

CNN Convolution Layer

Stride 1 with Padding



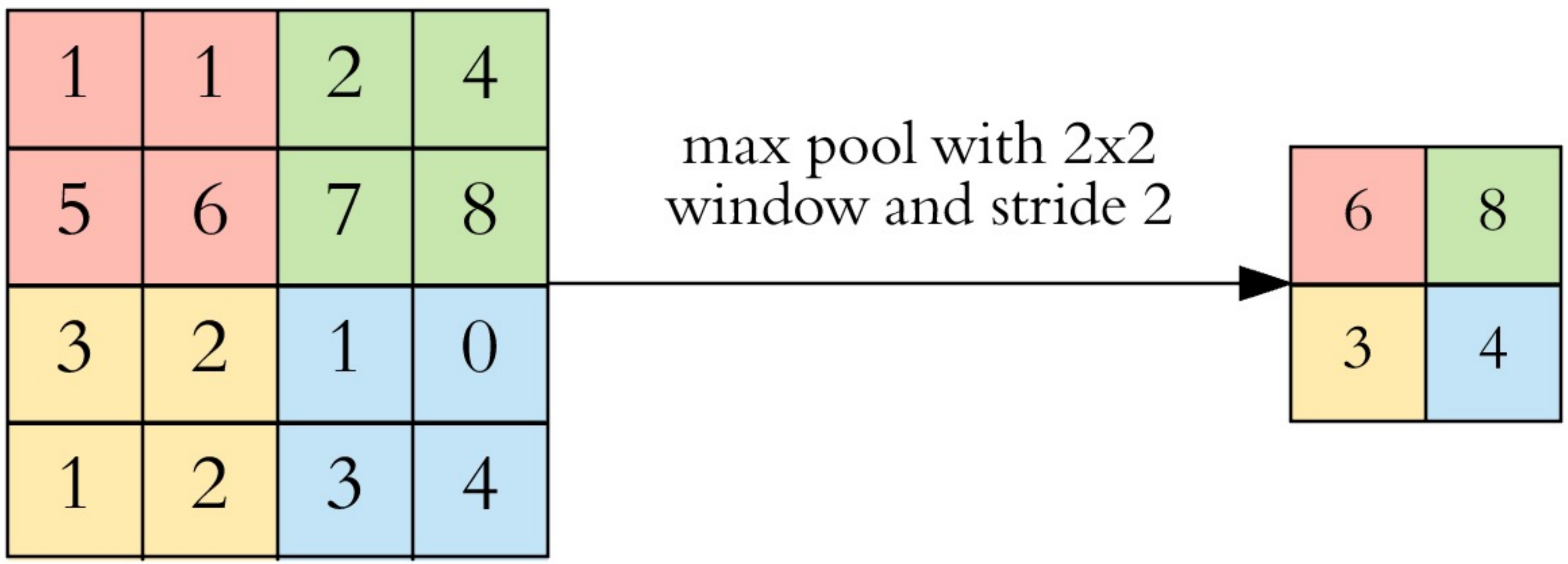
Stride 1 with Padding

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

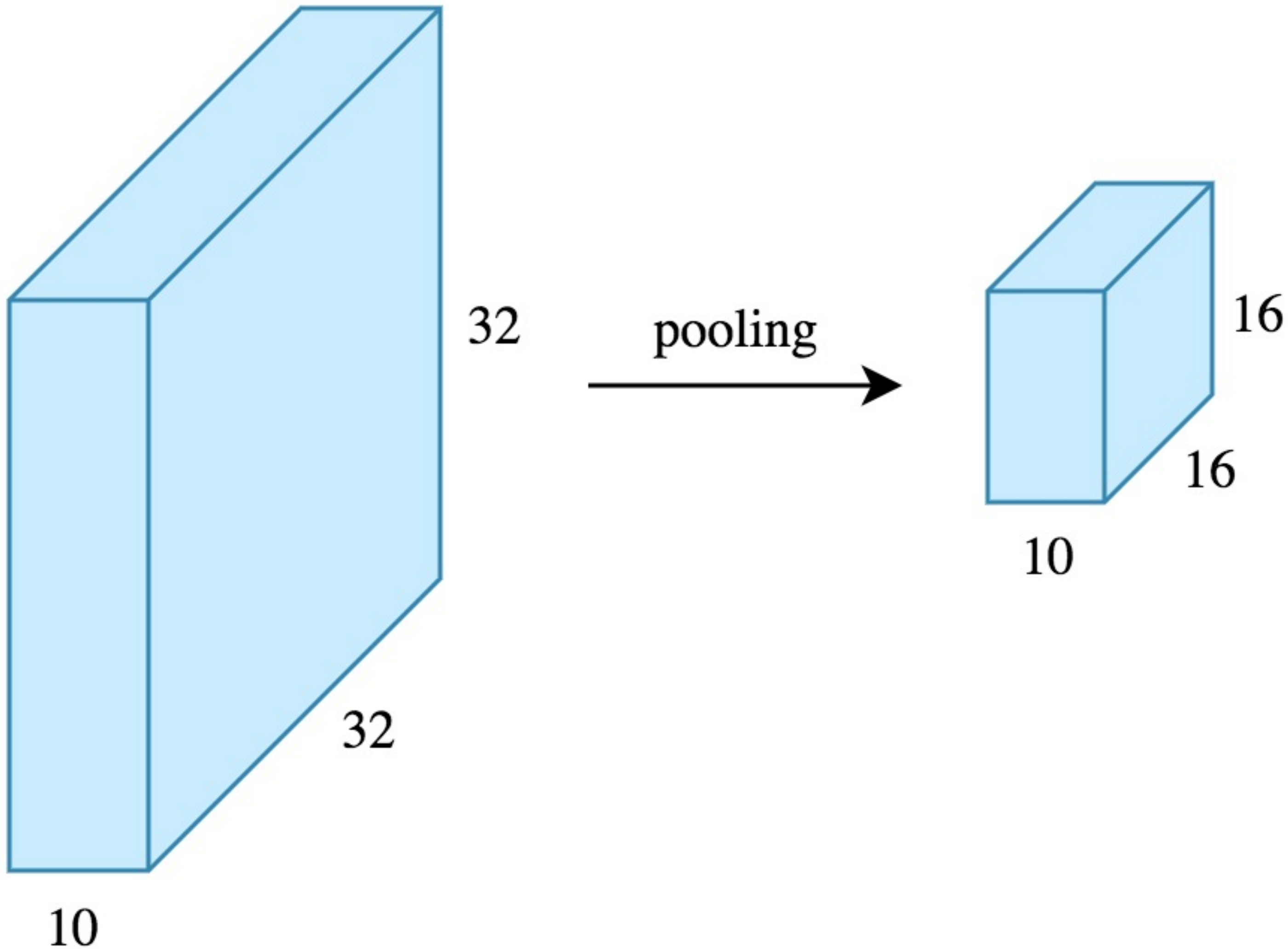
CNN Pooling Layer

Max Pooling



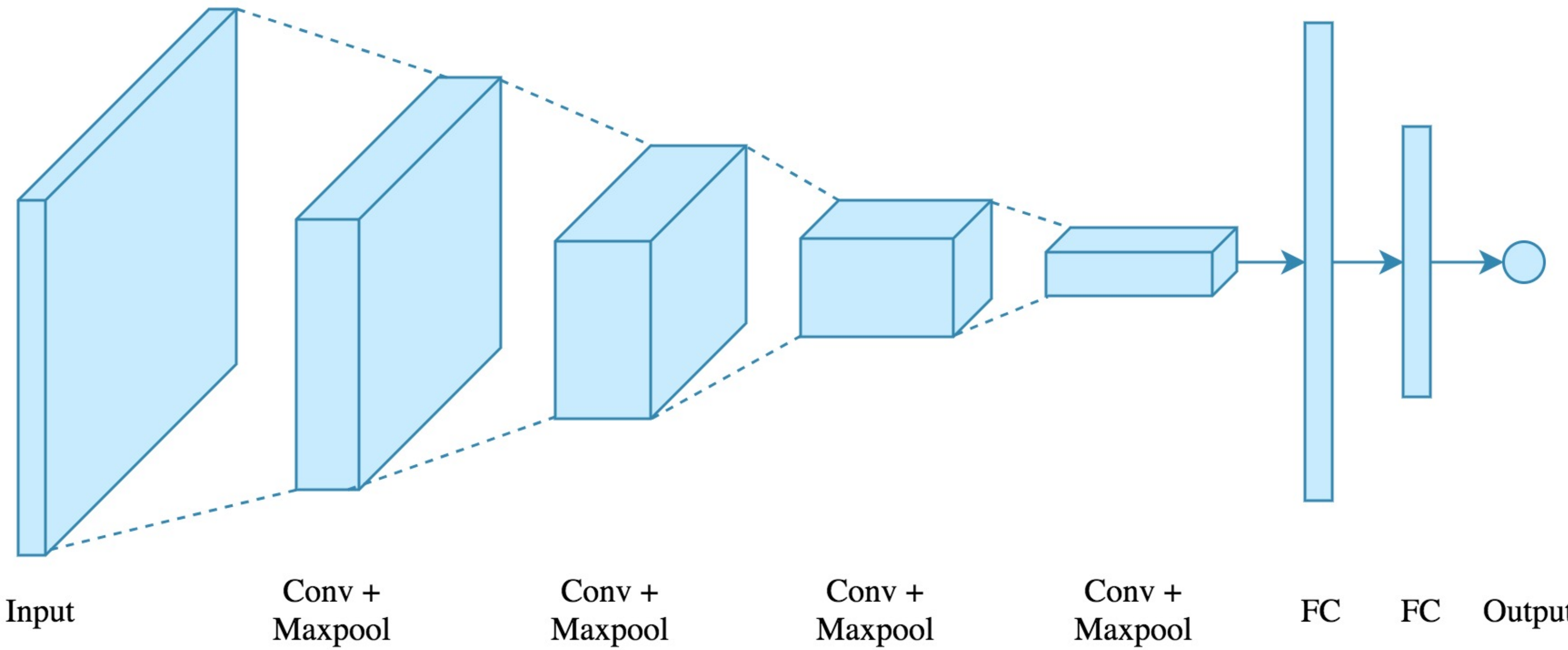
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks,
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

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

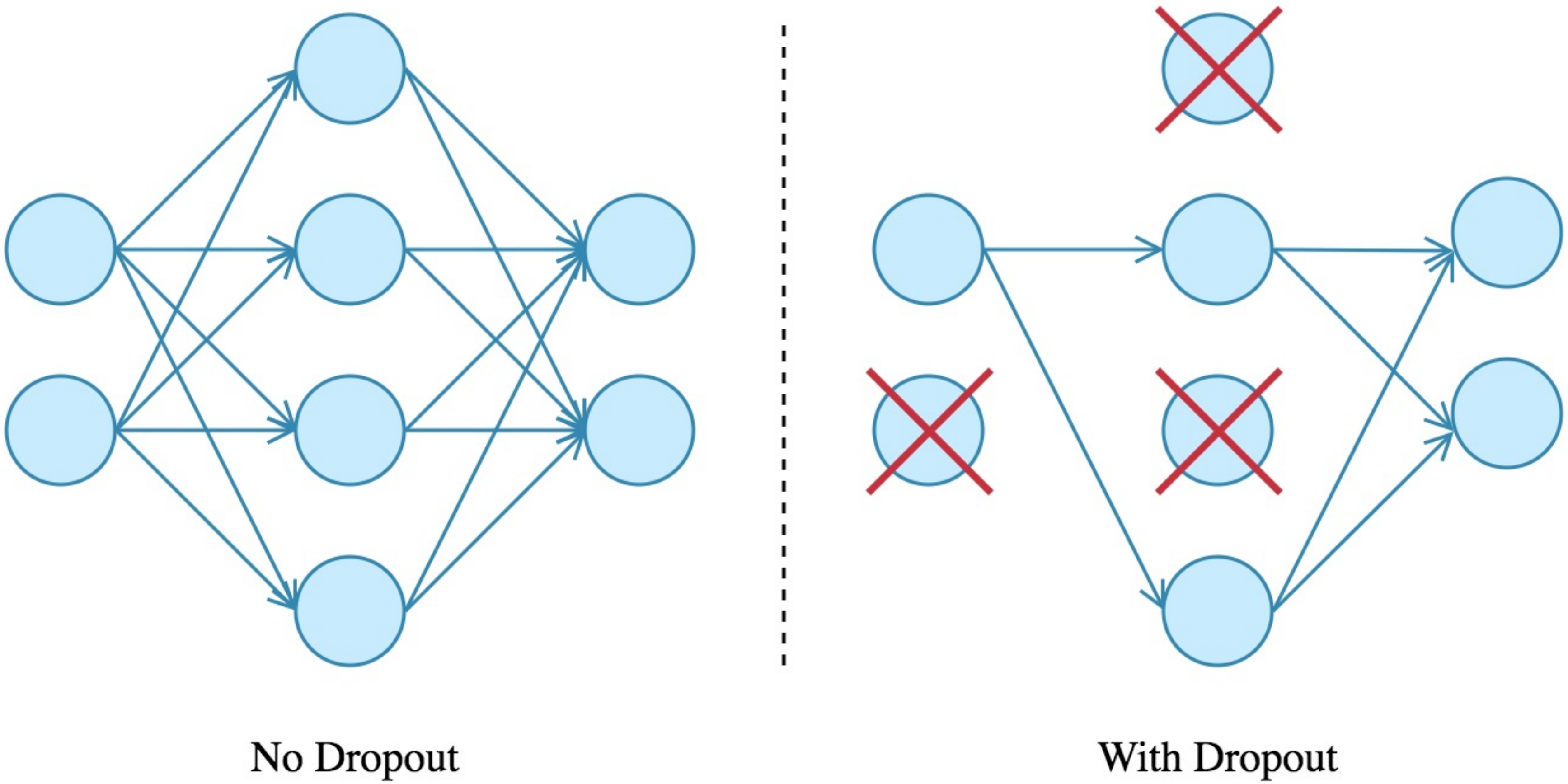
CNN Architecture
4 convolution + pooling layers, followed by 2 fully connected layers

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'))
```

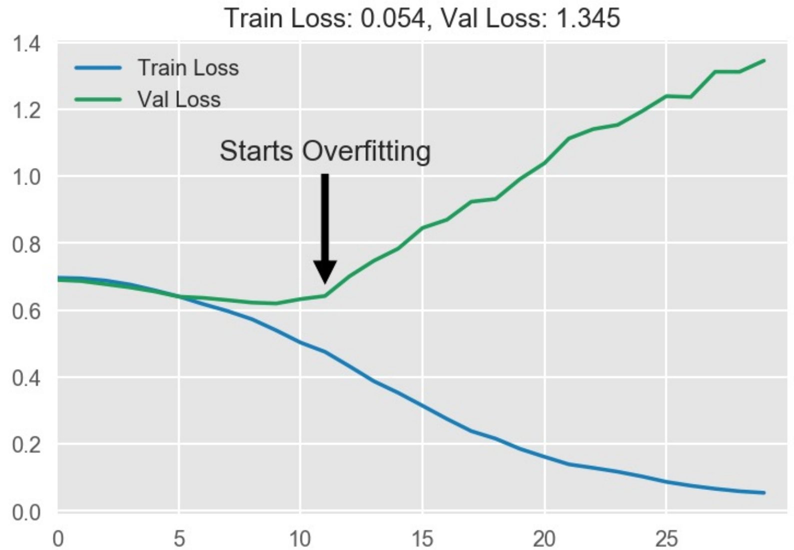
Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks,
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

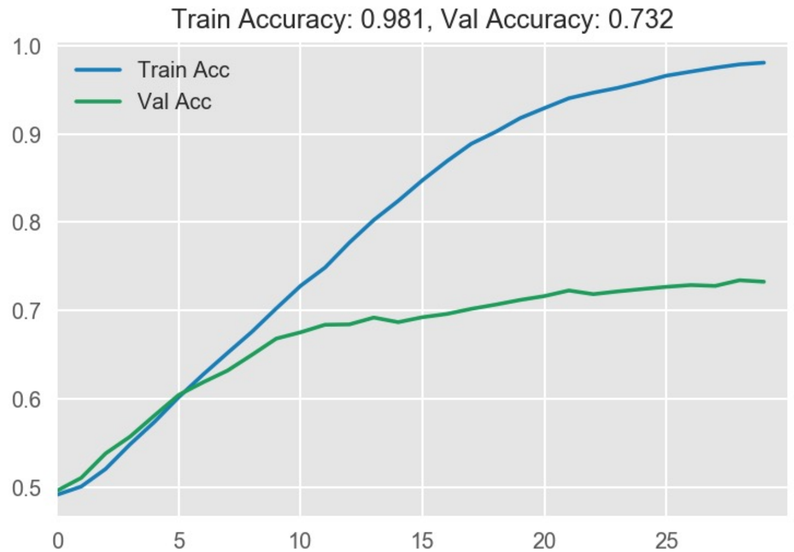
Dropout



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

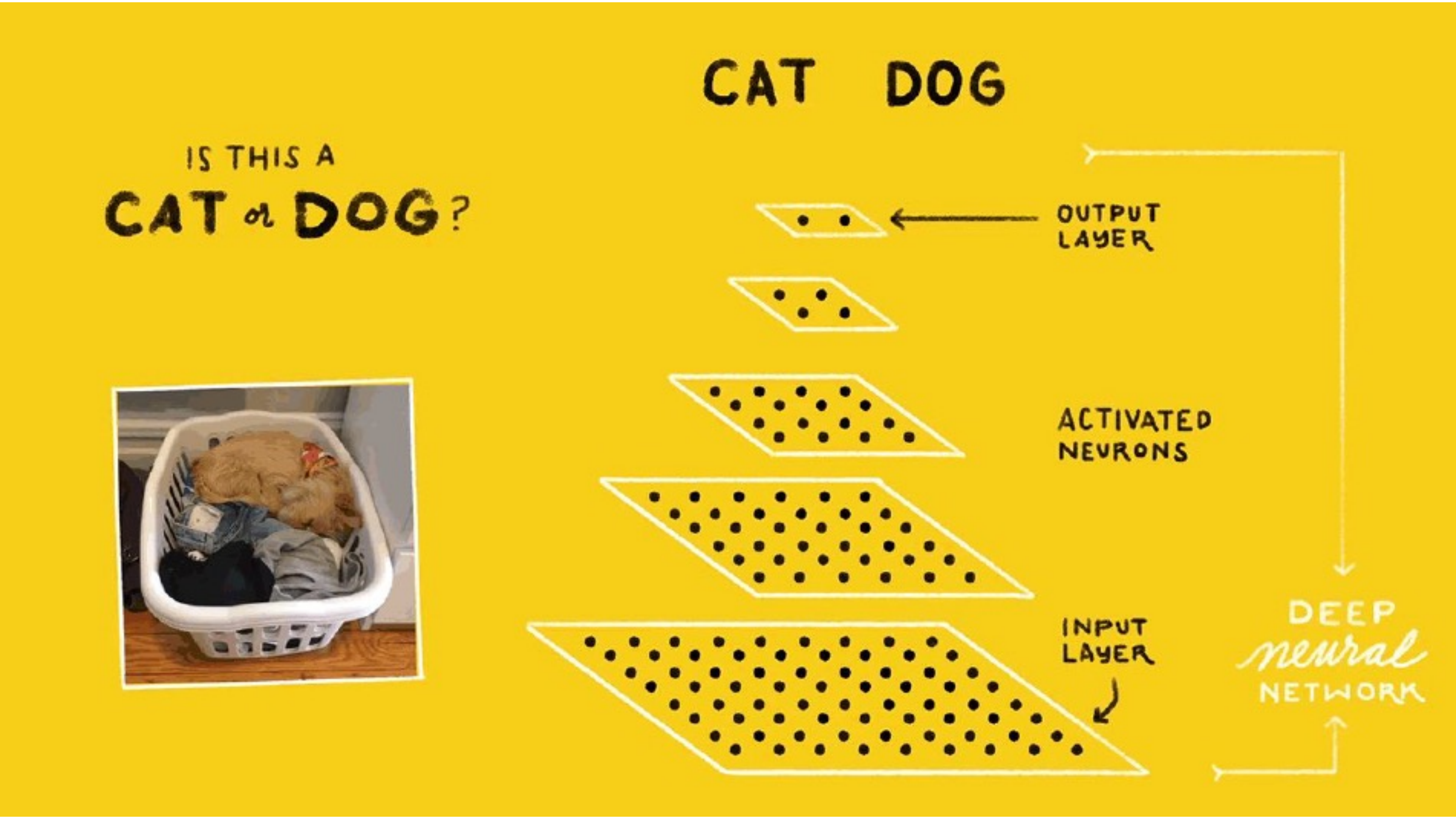




Source: Arden Dertat (2017), Applied Deep Learning - Part 4: Convolutional Neural Networks, https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Visual Recognition

Image Classification

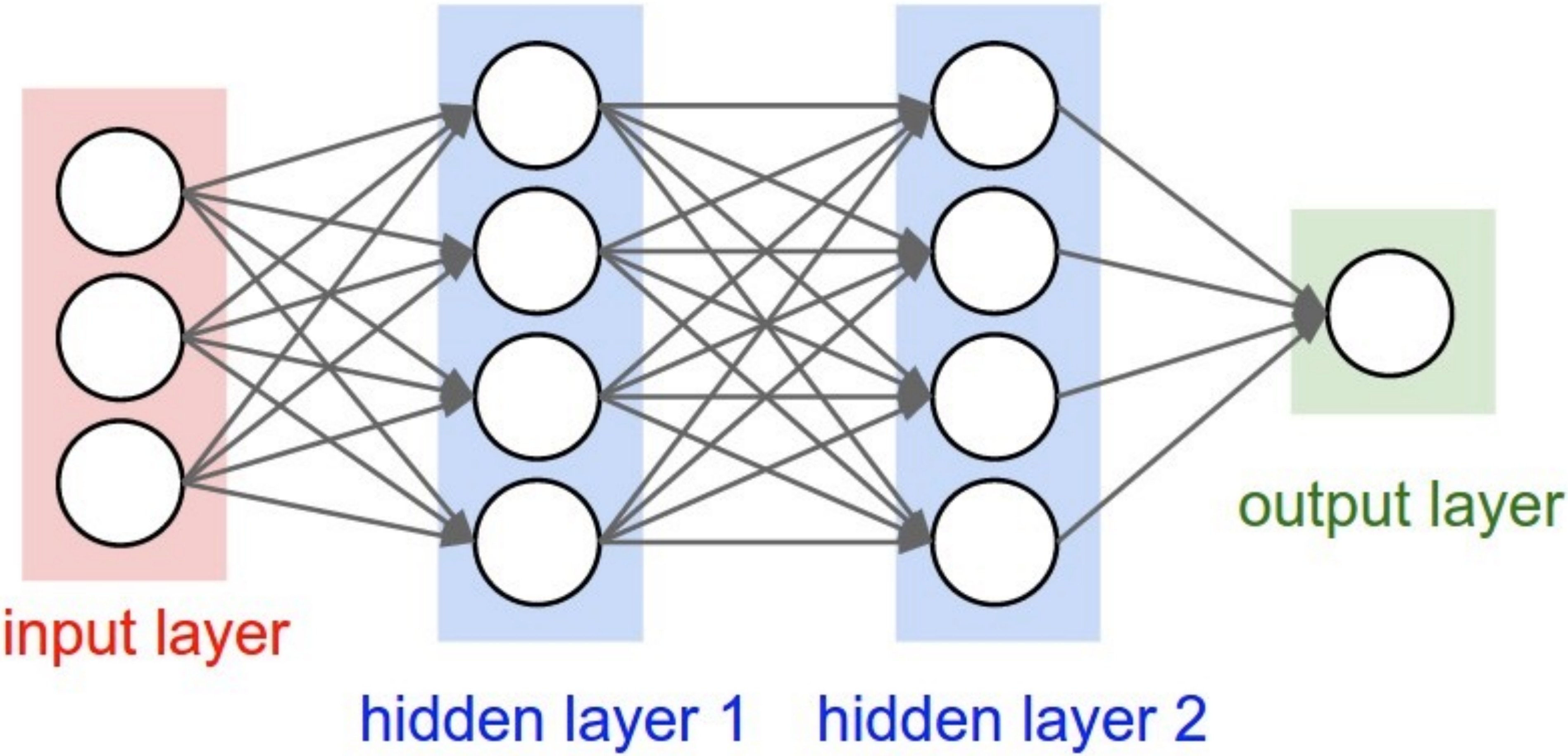


Source: Jeff Dean (2016), Large-Scale Deep Learning For Building Intelligent Computer Systems, WSDM 2016

Convolutional Neural Networks (CNNs / ConvNets)

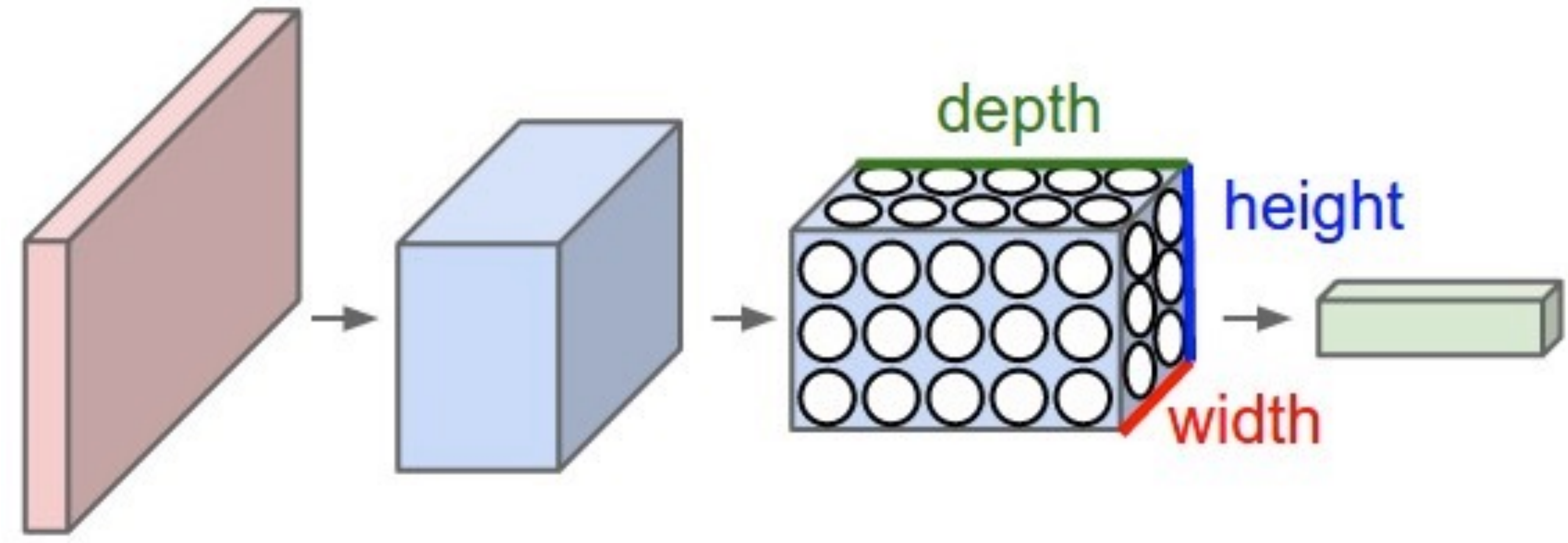
http://cs231n.github.io/convolutional-networks/

A regular 3-layer Neural Network



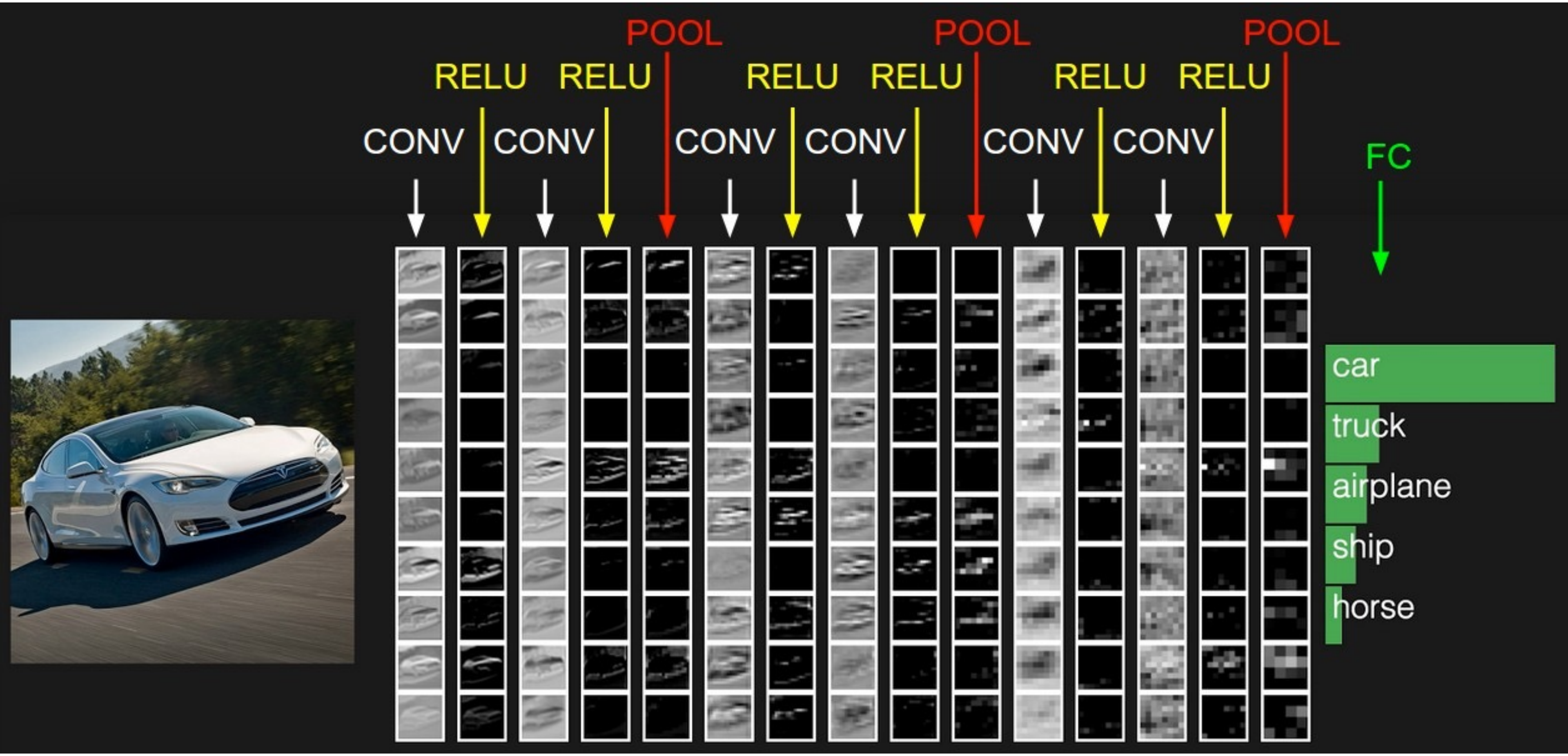
http://cs231n.github.io/convolutional-networks/

A ConvNet arranges its neurons in three dimensions (width, height, depth)



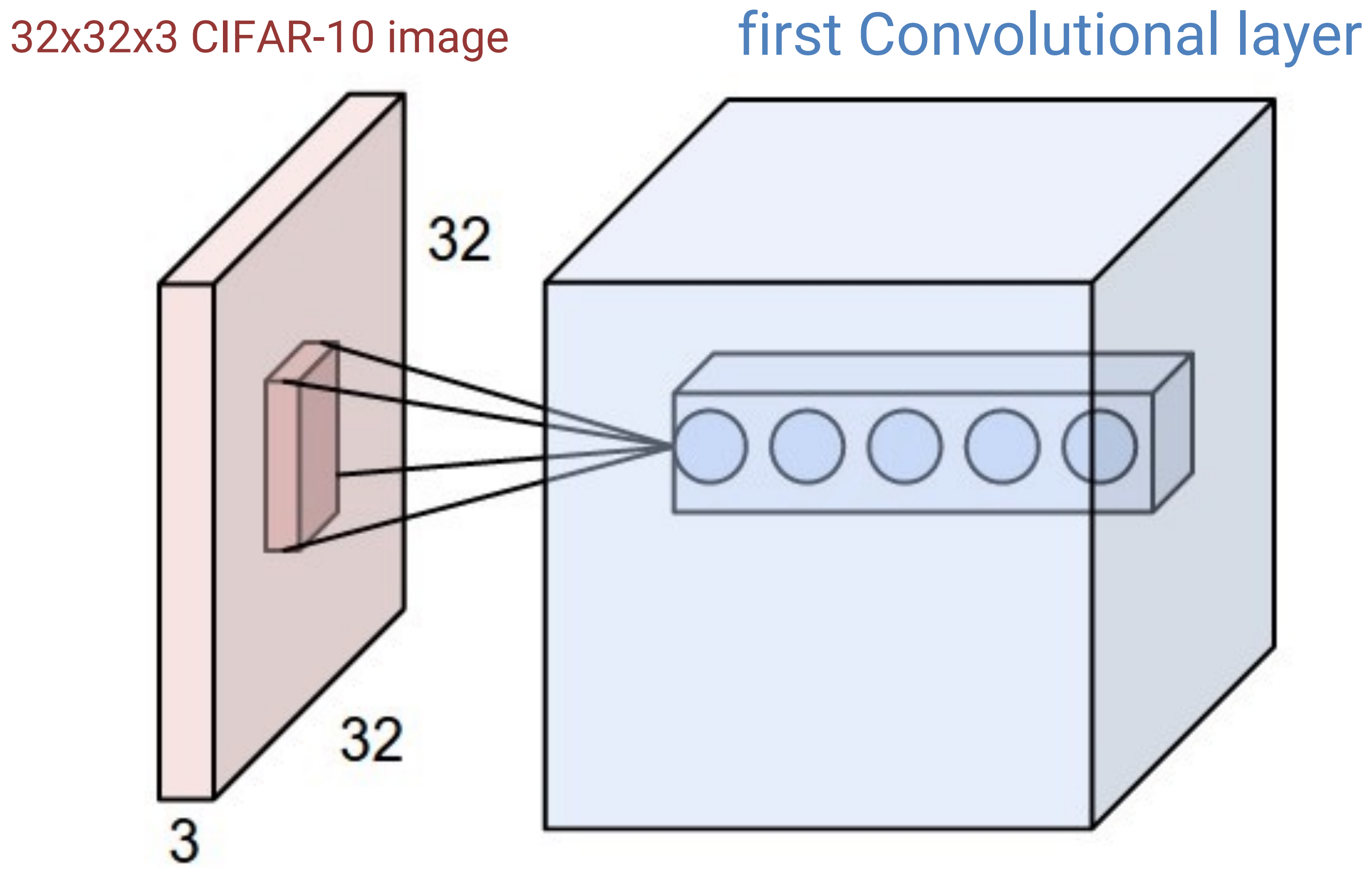
http://cs231n.github.io/convolutional-networks/

The activations of an example ConvNet architecture.



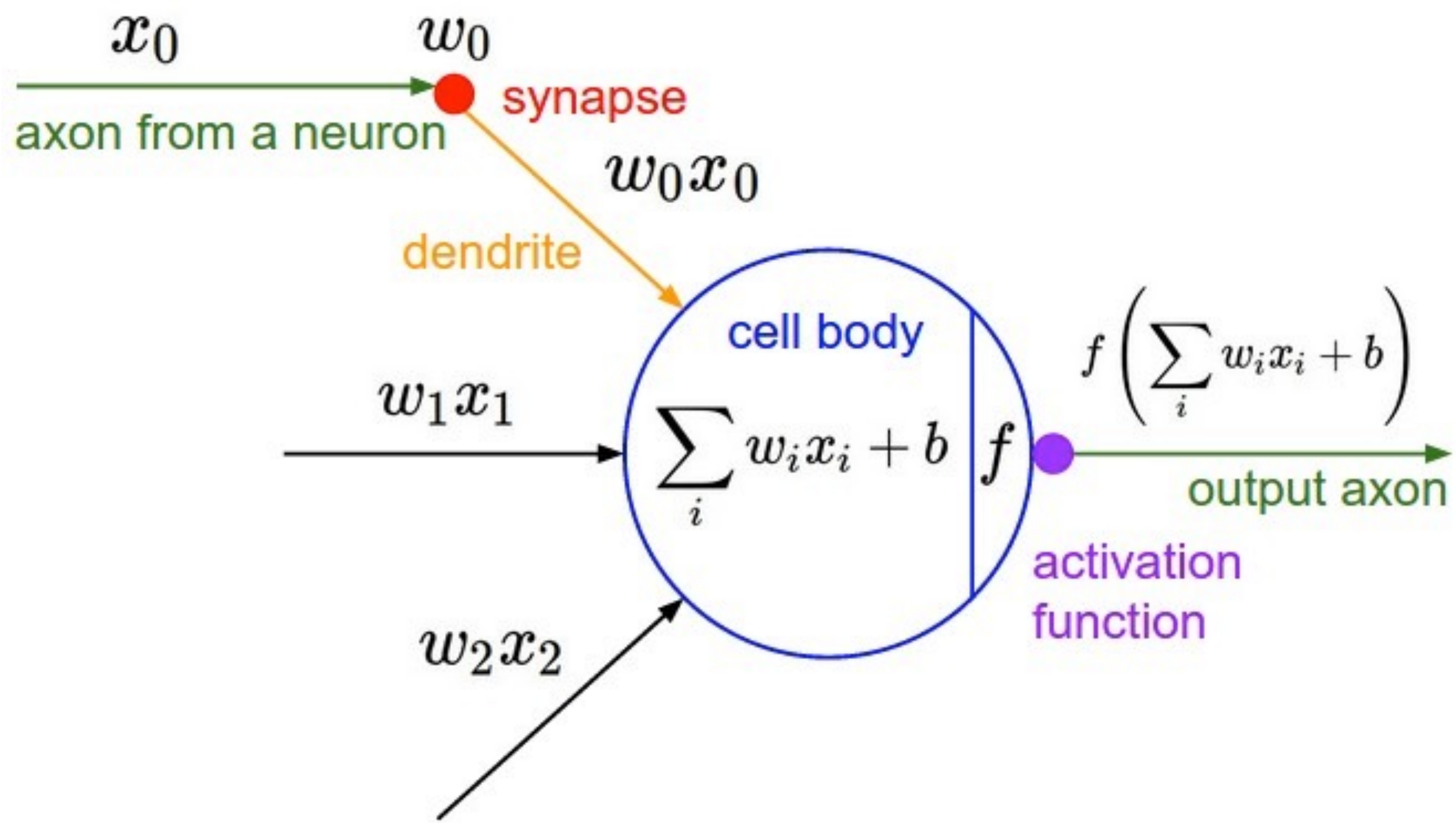
http://cs231n.github.io/convolutional-networks/

ConvNets



http://cs231n.github.io/convolutional-networks/

ConvNets



http://cs231n.github.io/convolutional-networks/

Convolution Demo

Input Volume (+pad 1) (7x7x3)

x[:,:,0]

0	0	0	0	0	0	0
0	1	2	0	2	1	0
0	2	2	2	1	1	0
0	2	2	2	0	1	0
0	2	2	1	2	1	0
0	2	1	2	0	1	0
0	0	0	0	0	0	0

x[:,:,1]

0	0	0	0	0	0	0
0	0	2	2	1	2	0
0	1	2	0	0	2	0
0	0	1	2	1	0	0
0	2	2	2	2	0	0
0	2	2	2	0	2	0
0	0	0	0	0	0	0

x[:,:,2]

0	0	0	0	0	0	0
0	1	0	0	1	0	0
0	0	2	0	0	0	0
0	0	0	1	1	1	0
0	2	2	2	1	2	0
0	1	2	0	0	2	0
0	0	0	0	0	0	0

Filter W0 (3x3x3)

w0[:,:,0]

-1	-1	0
1	1	1
-1	0	1

w0[:,:,1]

0	0	1
0	1	0
0	0	1

w0[:,:,2]

-1	-1	0
1	0	-1
-1	0	-1

Bias b0 (1x1x1)

b0[:,:,0]

1

Filter W1 (3x3x3)

w1[:,:,0]

1	-1	0
0	1	1
0	-1	1

w1[:,:,1]

-1	1	0
-1	-1	1
0	0	0

w1[:,:,2]

1	0	-1
0	0	-1
1	0	1

Bias b1 (1x1x1)

b1[:,:,0]

0

Output Volume (3x3x2)

o[:,:,0]

6	3	6
7	-1	-2
2	3	-2

o[:,:,1]

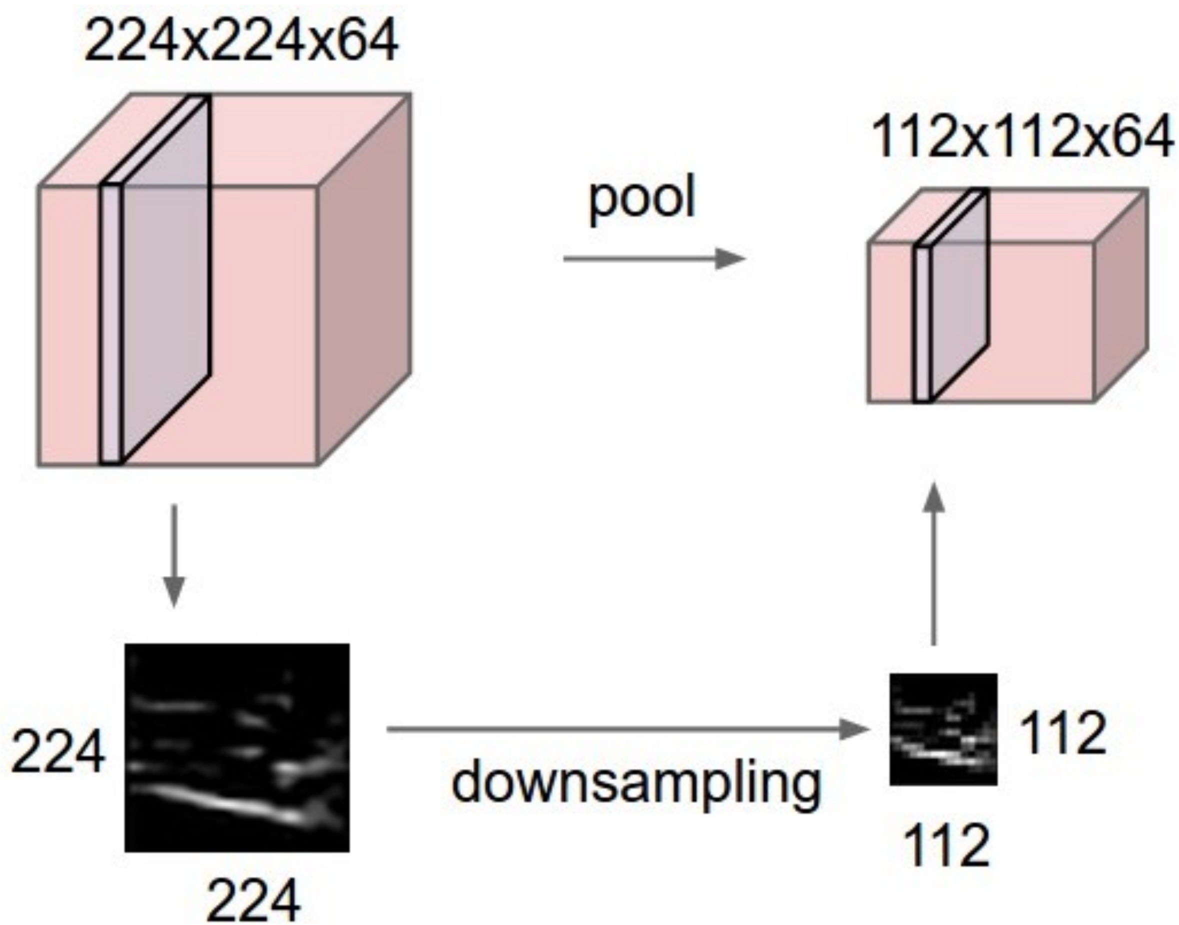
7	-1	-3
4	3	2
-1	0	-1

toggle movement

http://cs231n.github.io/convolutional-networks/

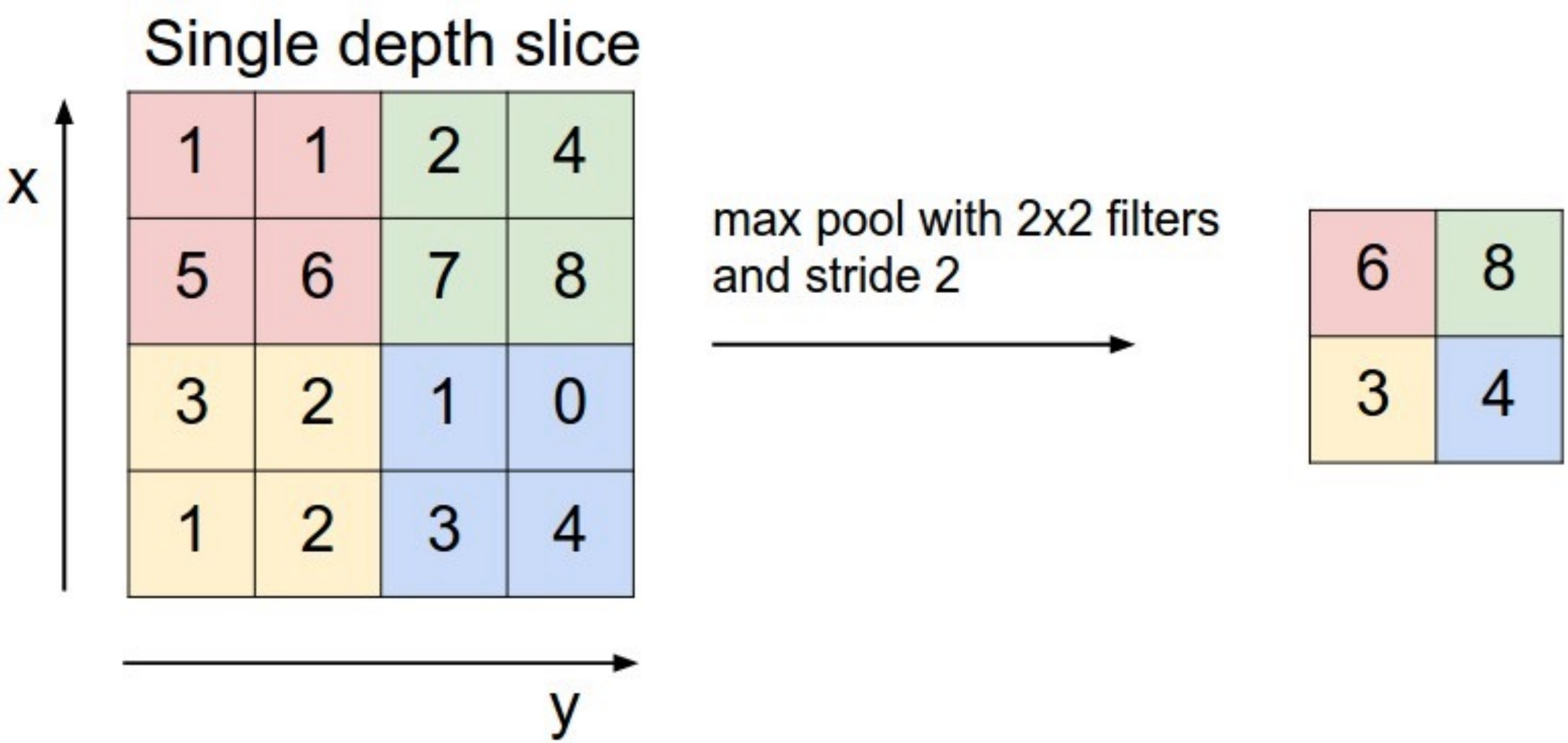
ConvNets

input volume of size [224x224x64]
is pooled with **filter** size 2, **stride** 2
into output volume of size [112x112x64]



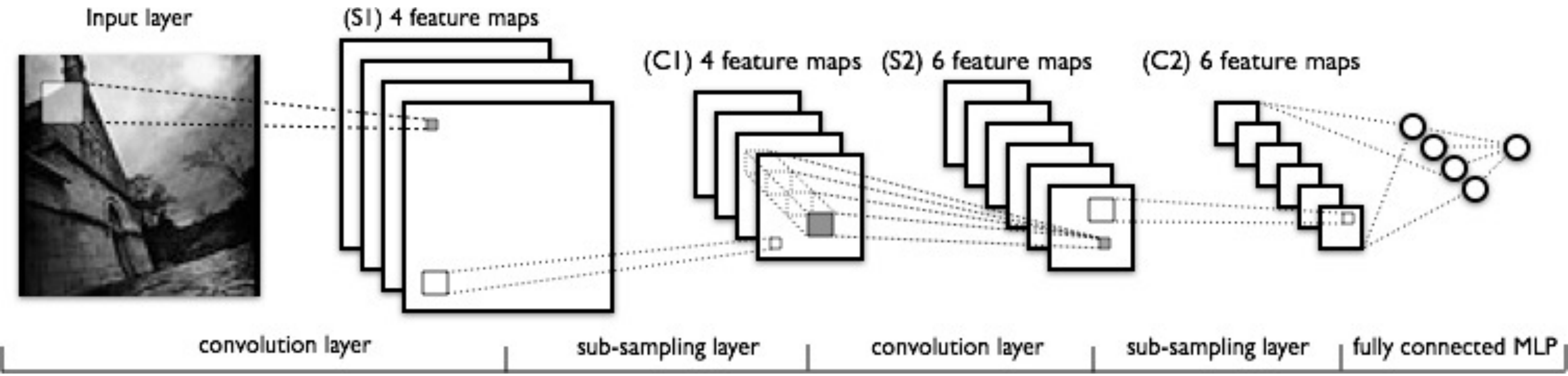
http://cs231n.github.io/convolutional-networks/

ConvNets max pooling



http://cs231n.github.io/convolutional-networks/

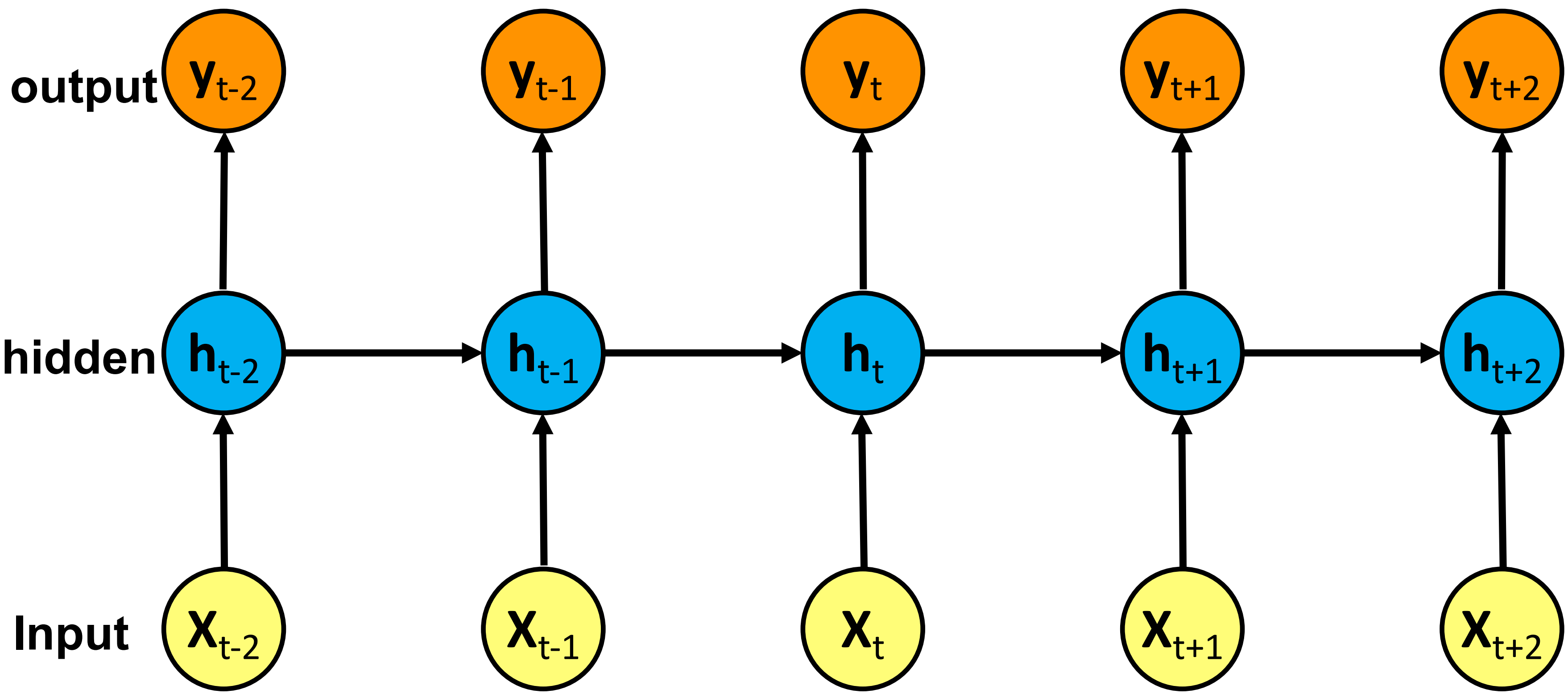
Convolutional Neural Networks (CNN) (LeNet)



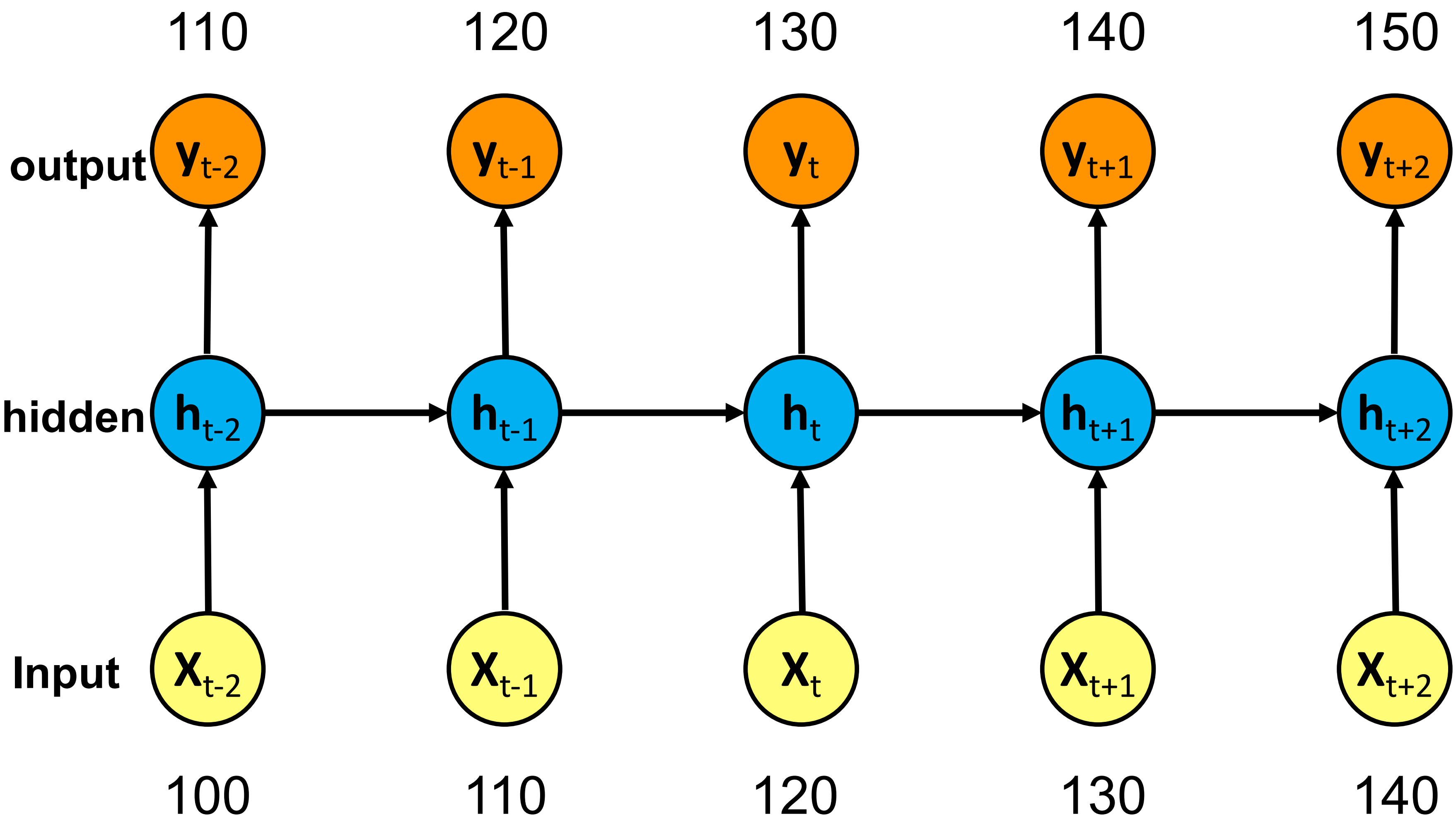
Source: http://deeplearning.net/tutorial/lenet.html

Recurrent Neural Networks (RNN)

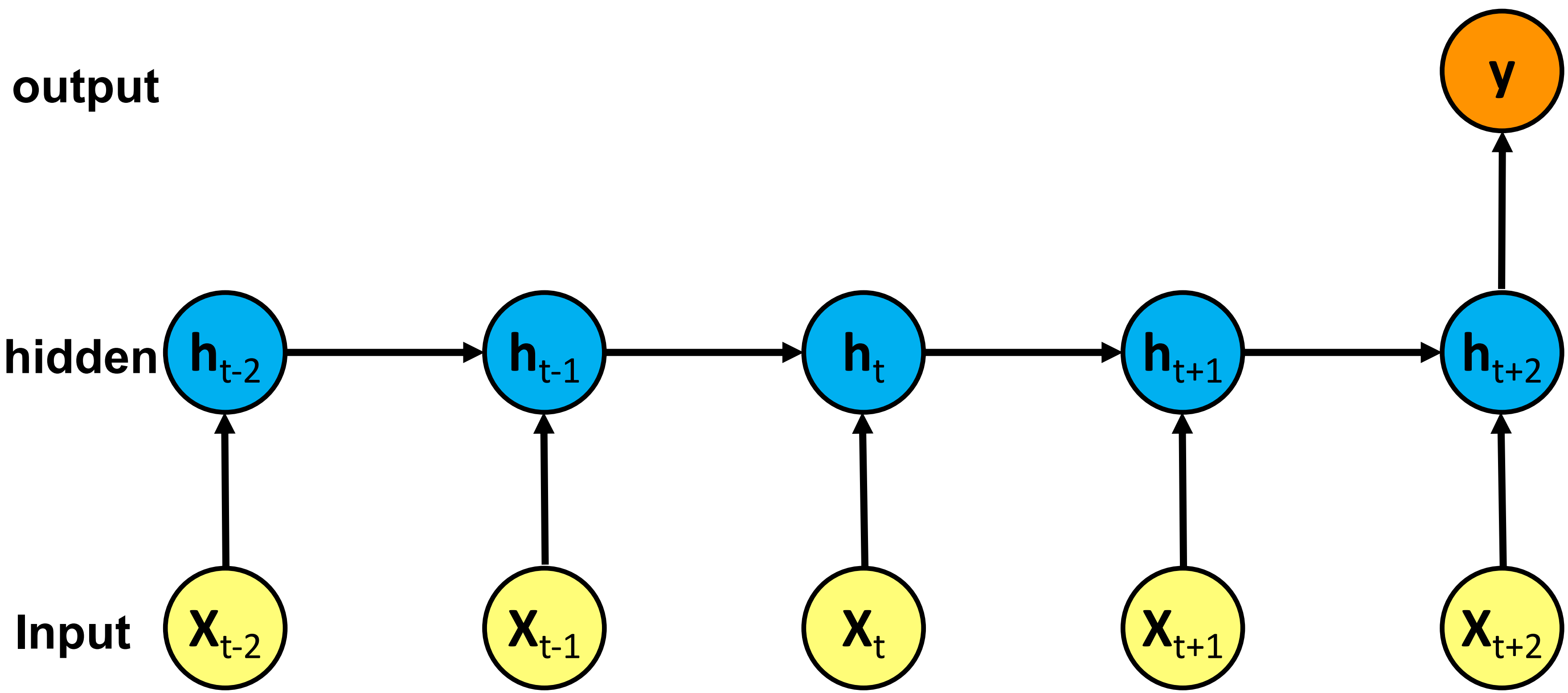
Recurrent Neural Networks (RNN)



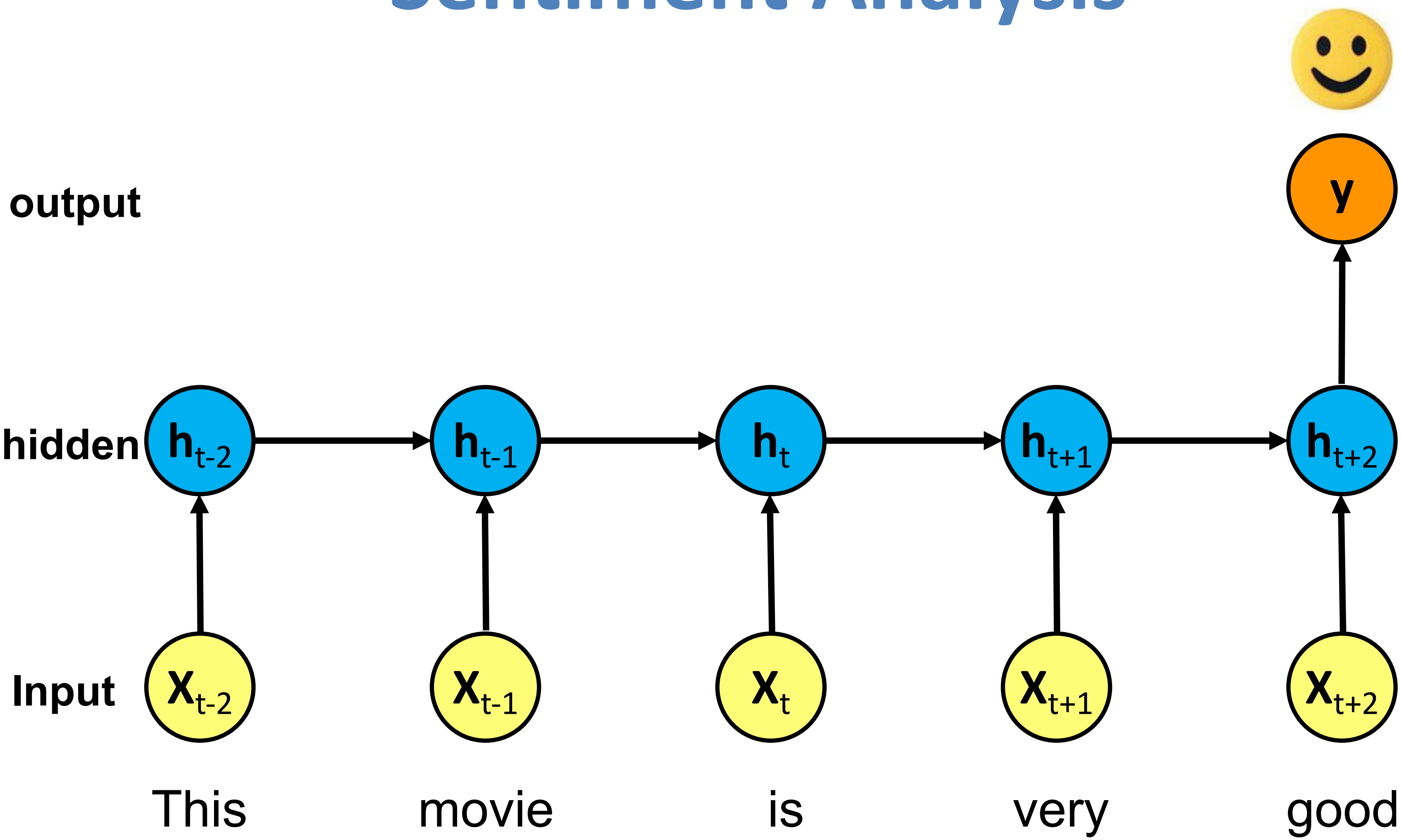
Recurrent Neural Networks (RNN) Time Series Forecasting



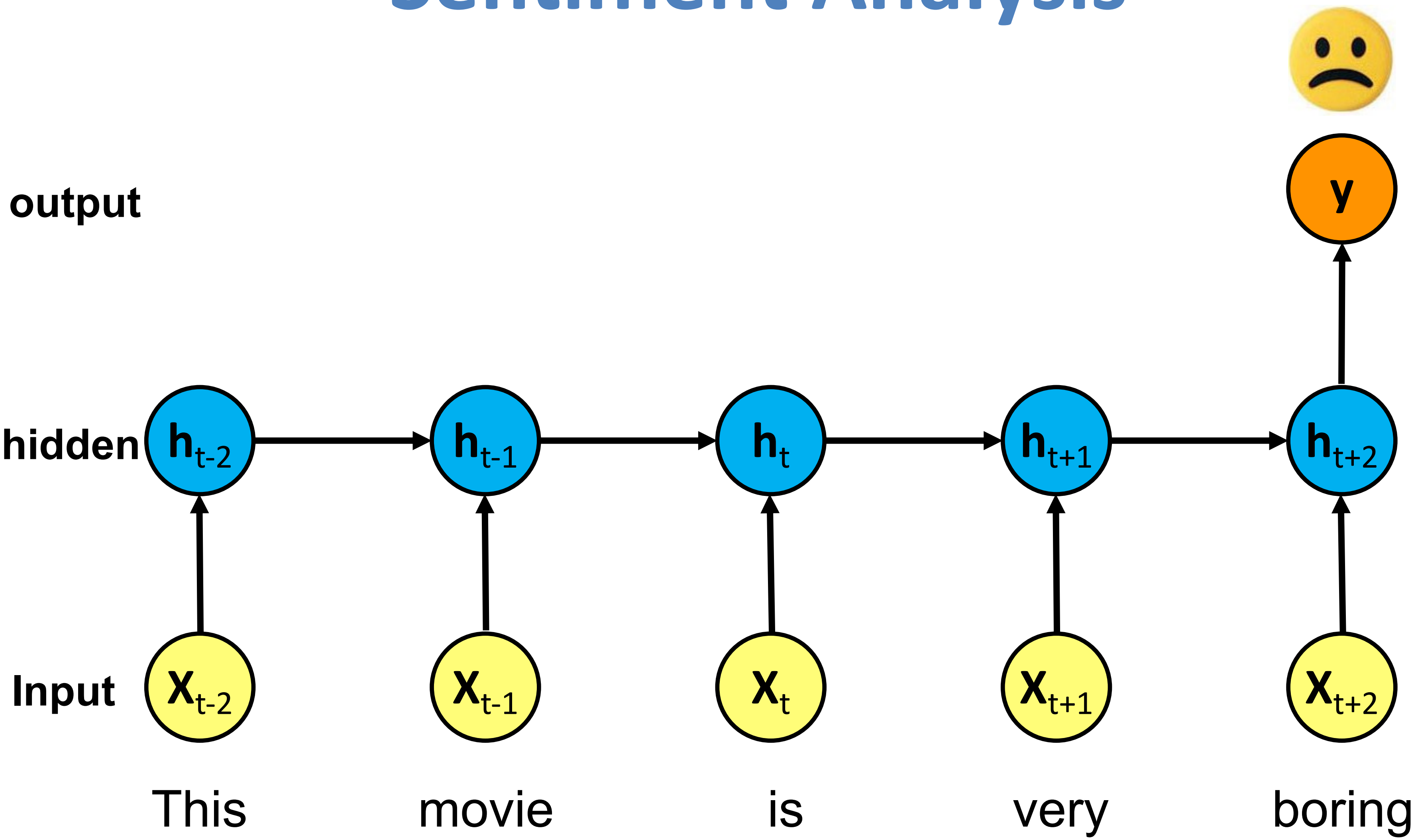
Recurrent Neural Networks (RNN)



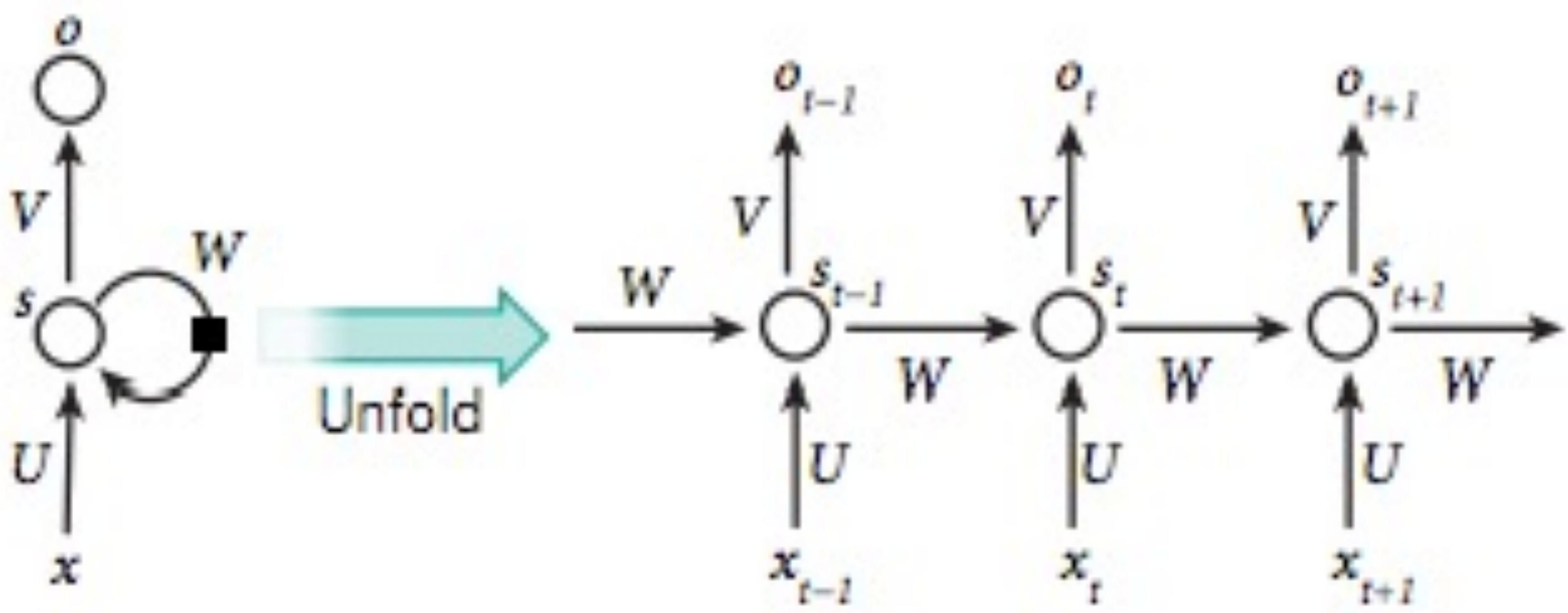
Recurrent Neural Networks (RNN) Sentiment Analysis



Recurrent Neural Networks (RNN)
Sentiment Analysis

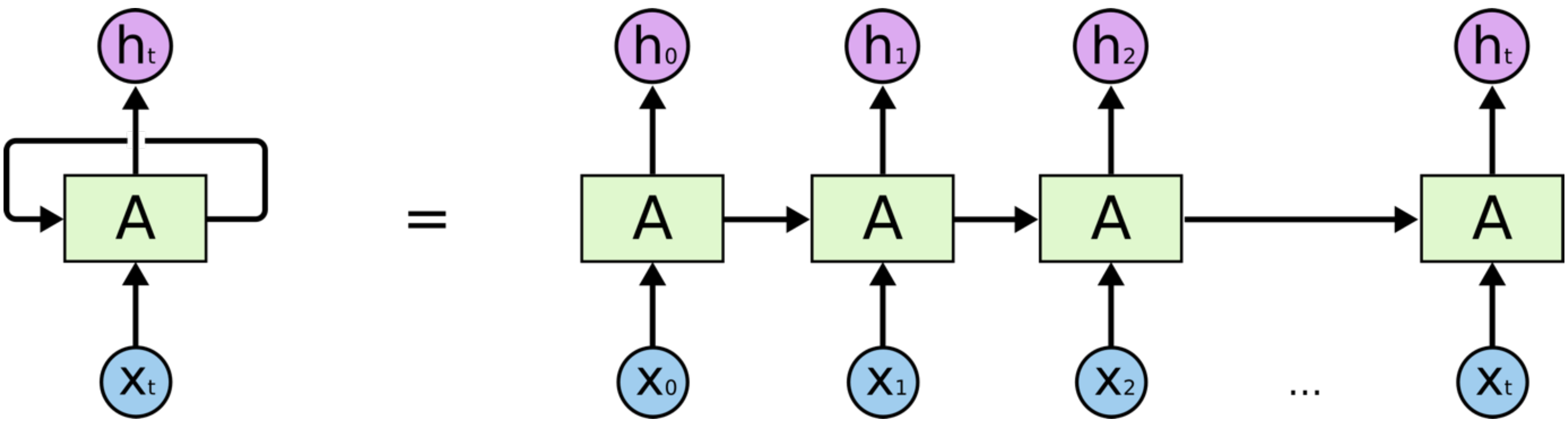


Recurrent Neural Network (RNN)



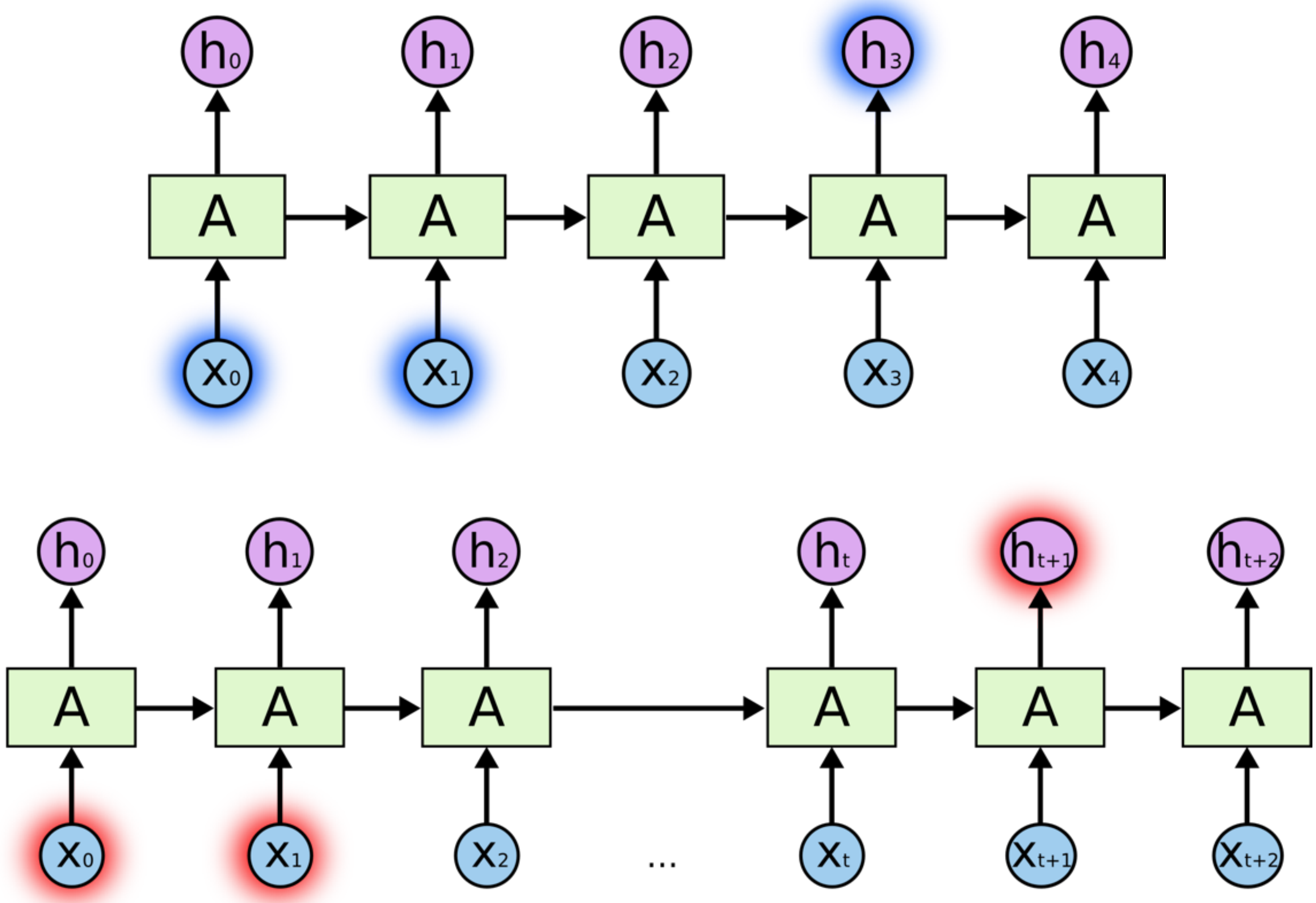
Source: LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521, no. 7553 (2015): 436-444.

RNN



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

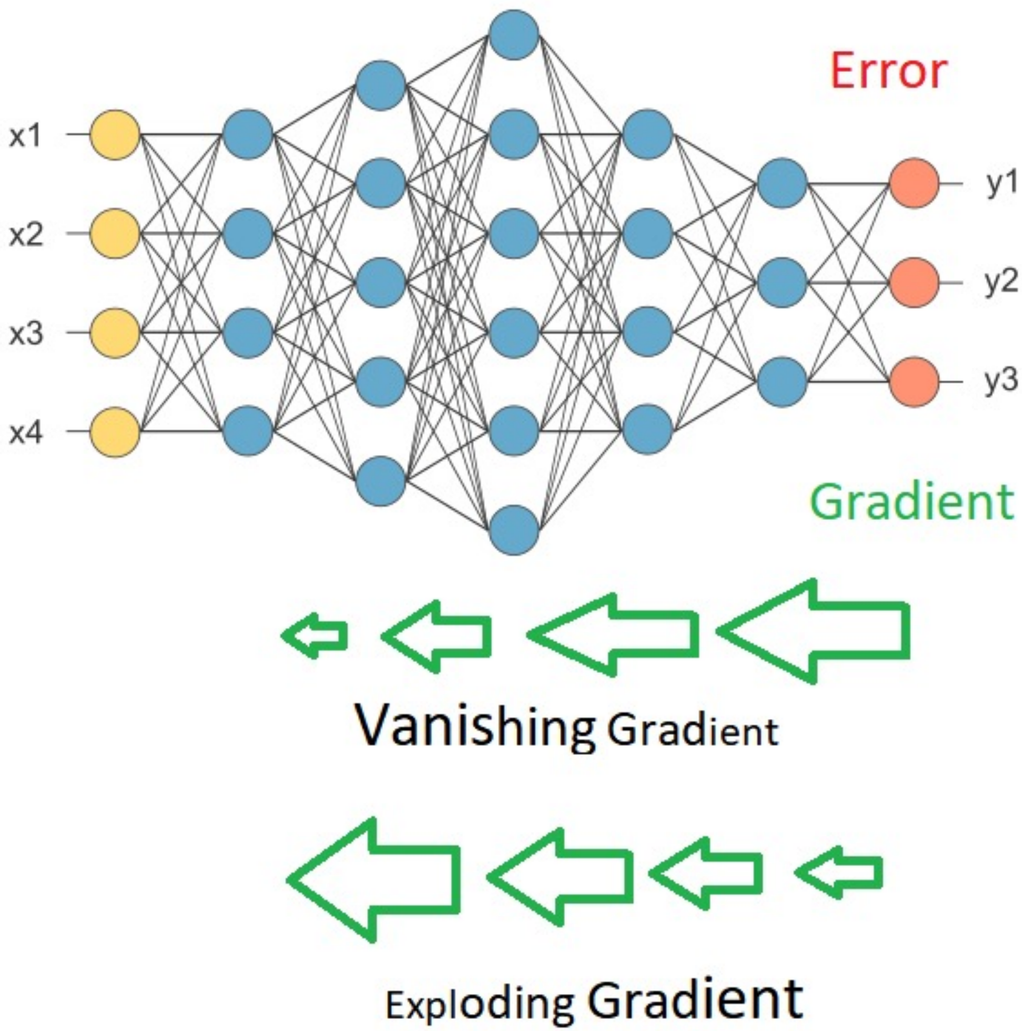
RNN long-term dependencies



I grew up in France… I speak fluent French.

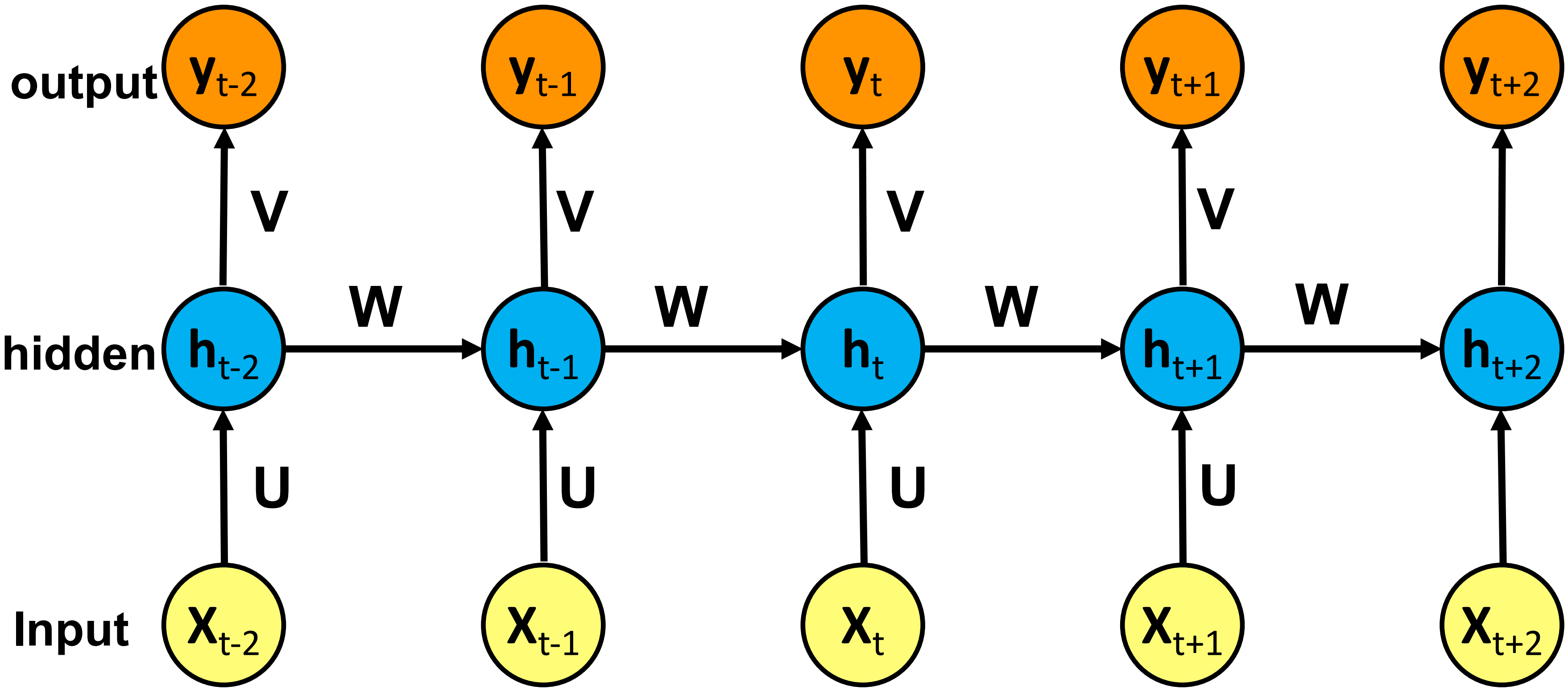
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Vanishing Gradient
Exploding Gradient

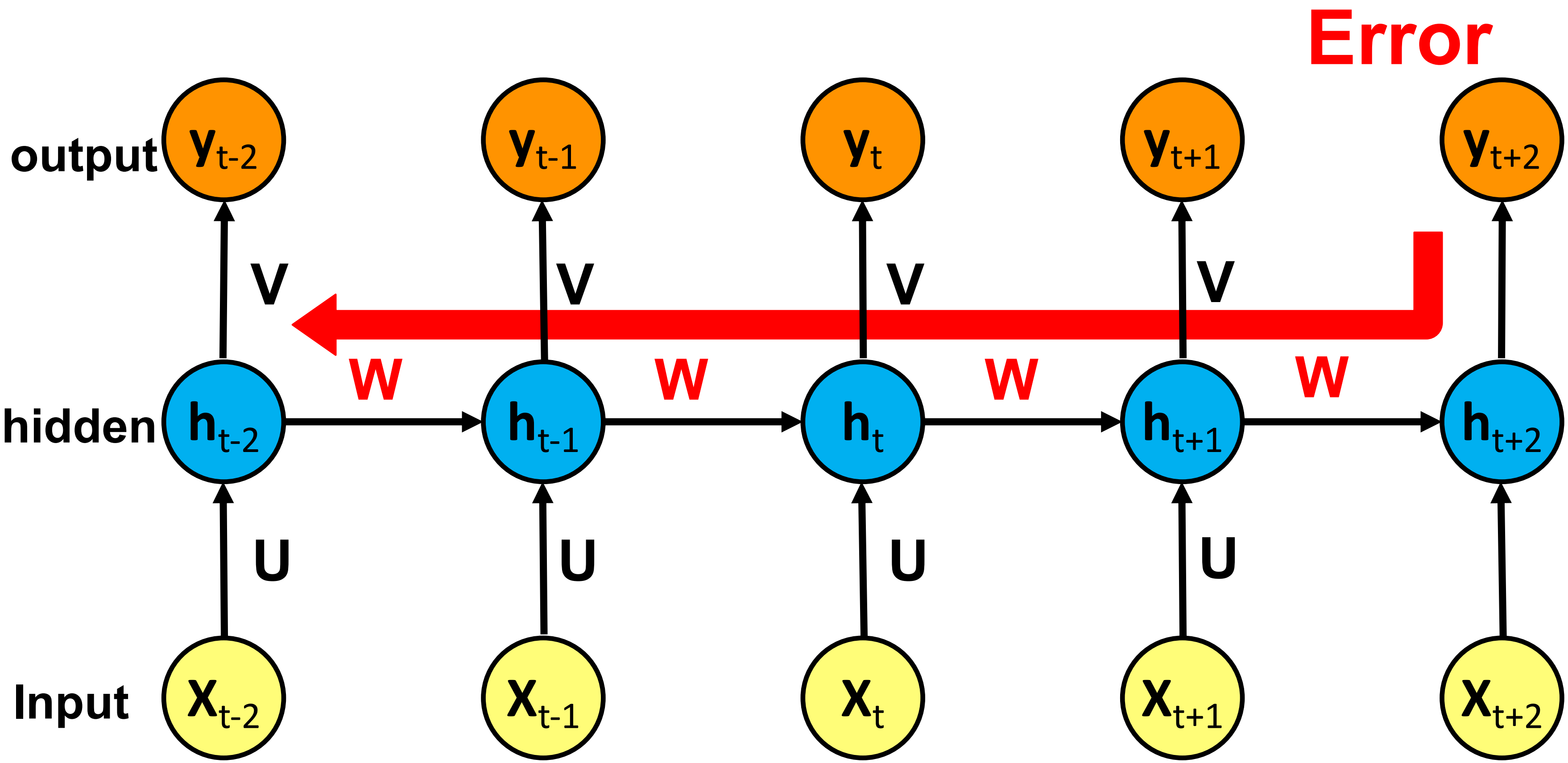


Source: https://medium.com/deep-math-machine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235

Recurrent Neural Networks (RNN)



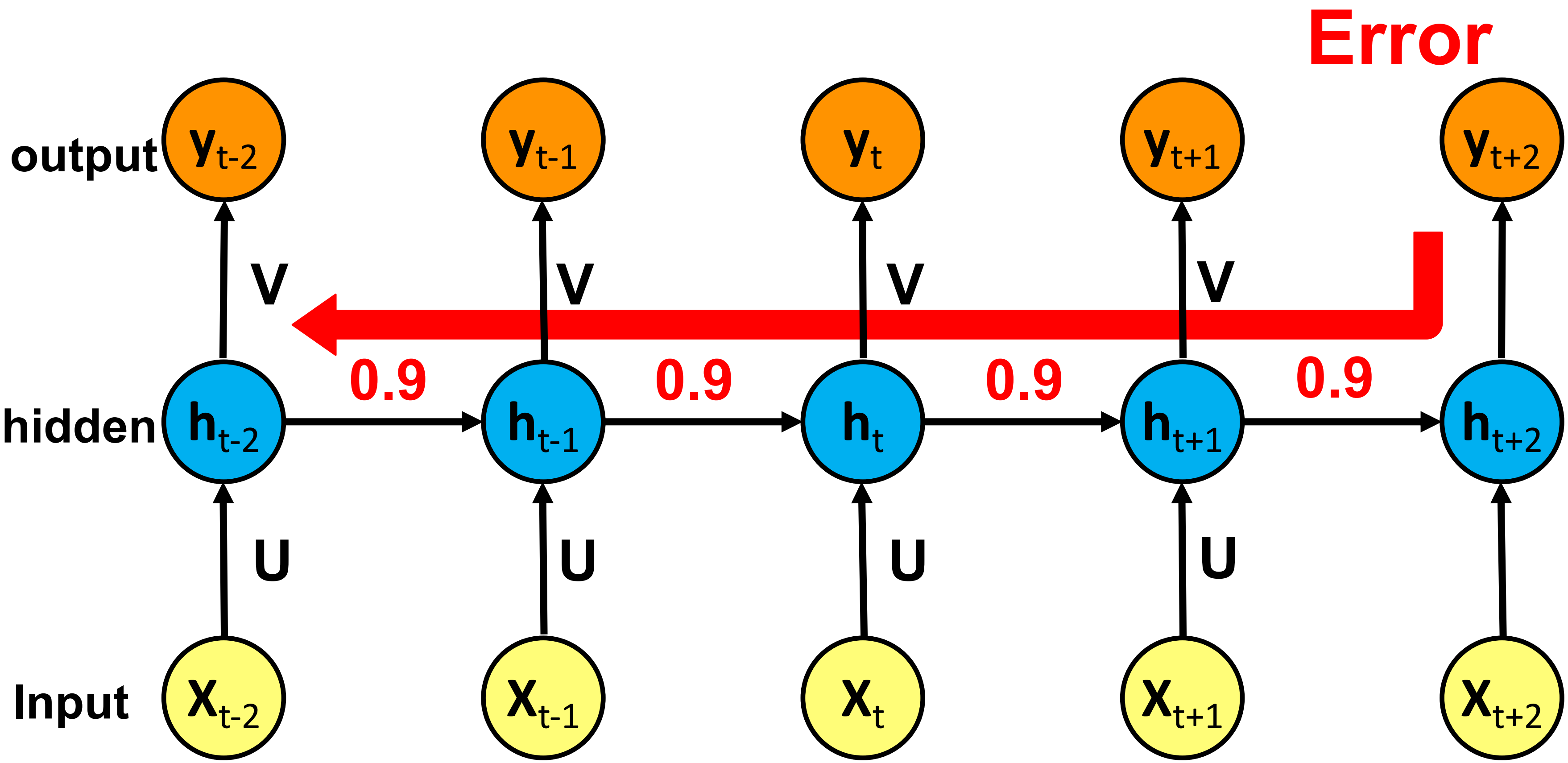
RNN
Vanishing Gradient problem
Exploding Gradient problem



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

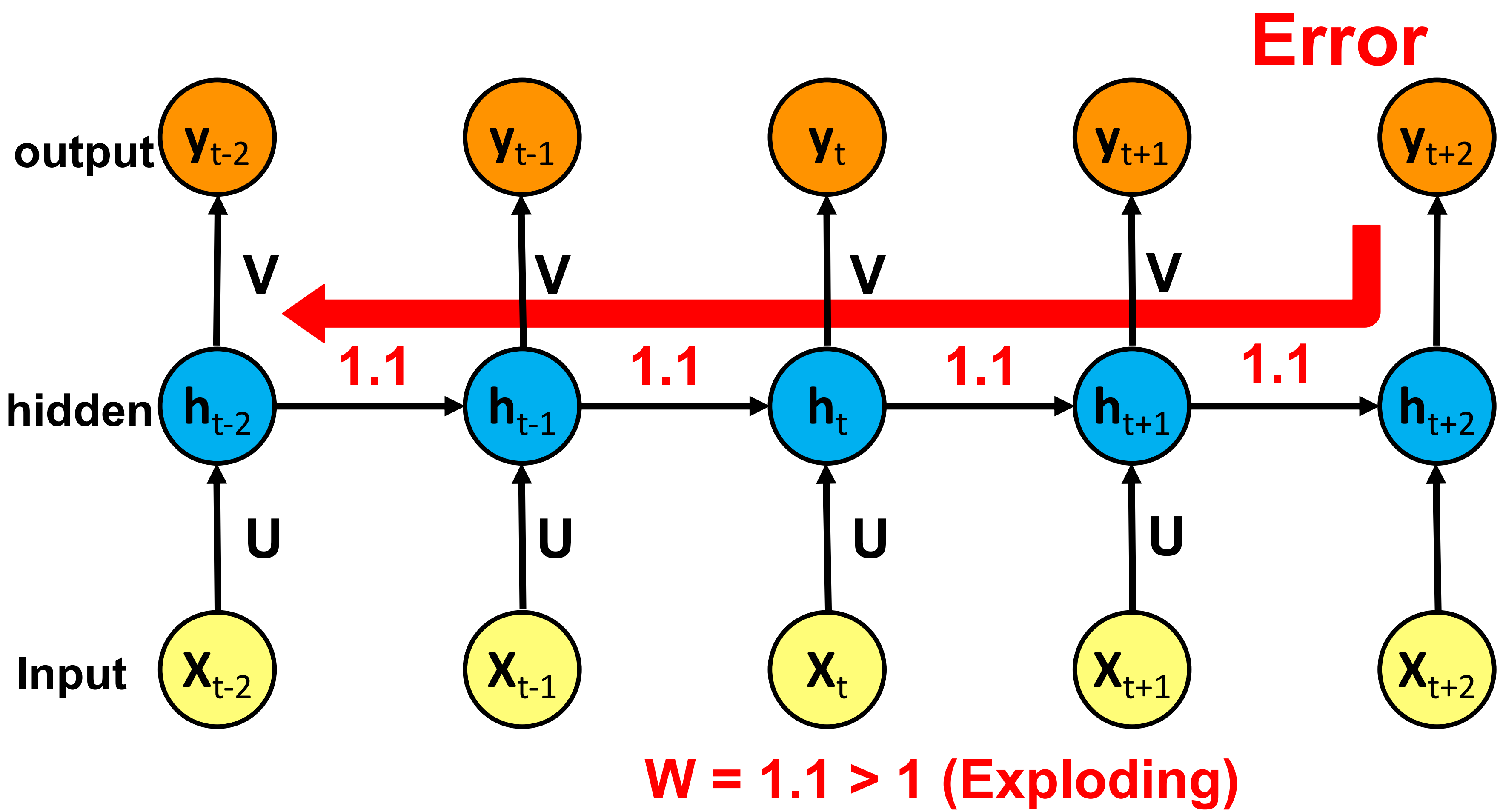
RNN
Vanishing Gradient problem



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

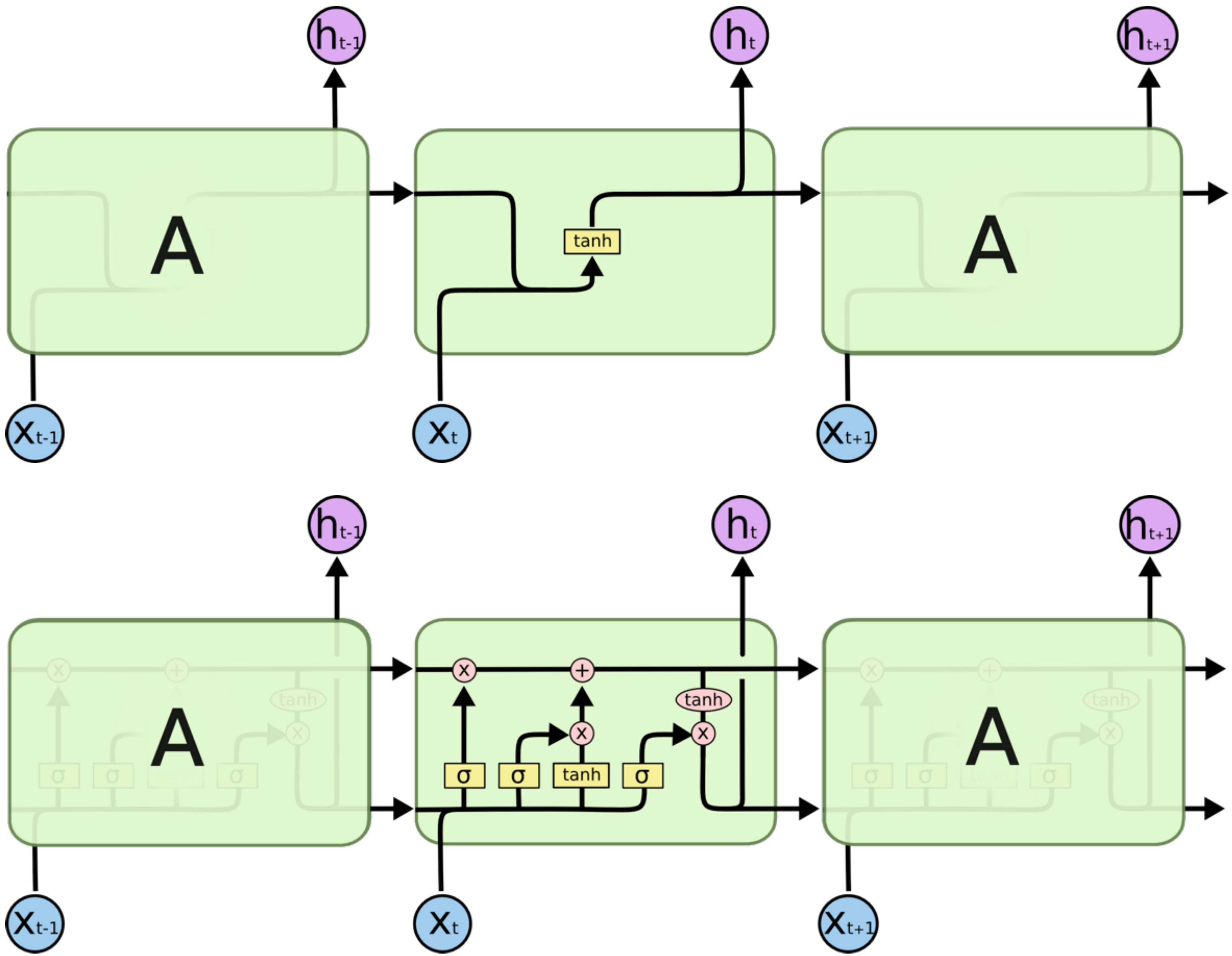


Source: https://medium.com/deep-math-machine-learning-ai/chapter-10-1-deepnlp-lstm-long-short-term-memory-networks-with-math-21477f8e4235

RNN LSTM

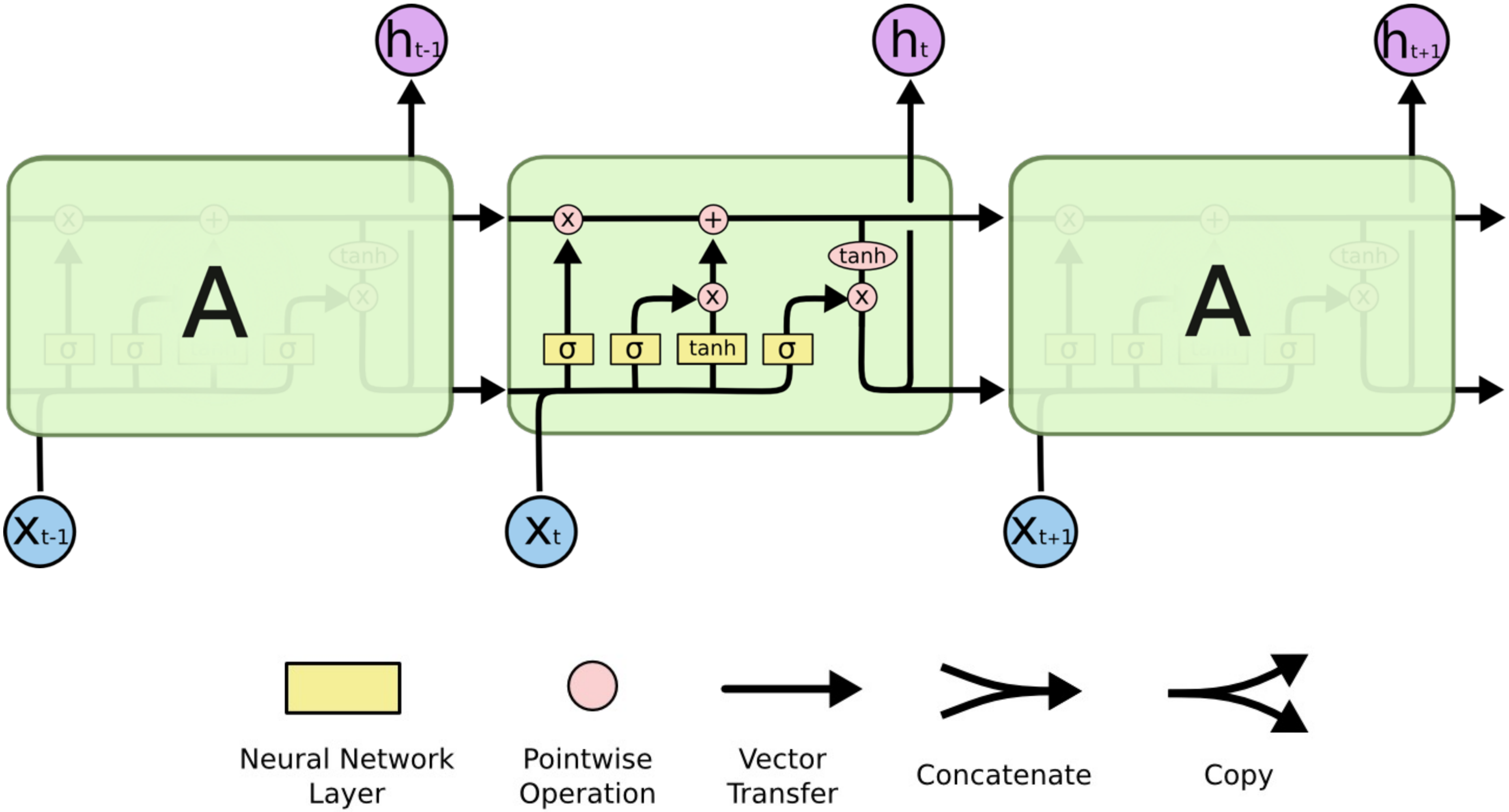
RNN

LSTM



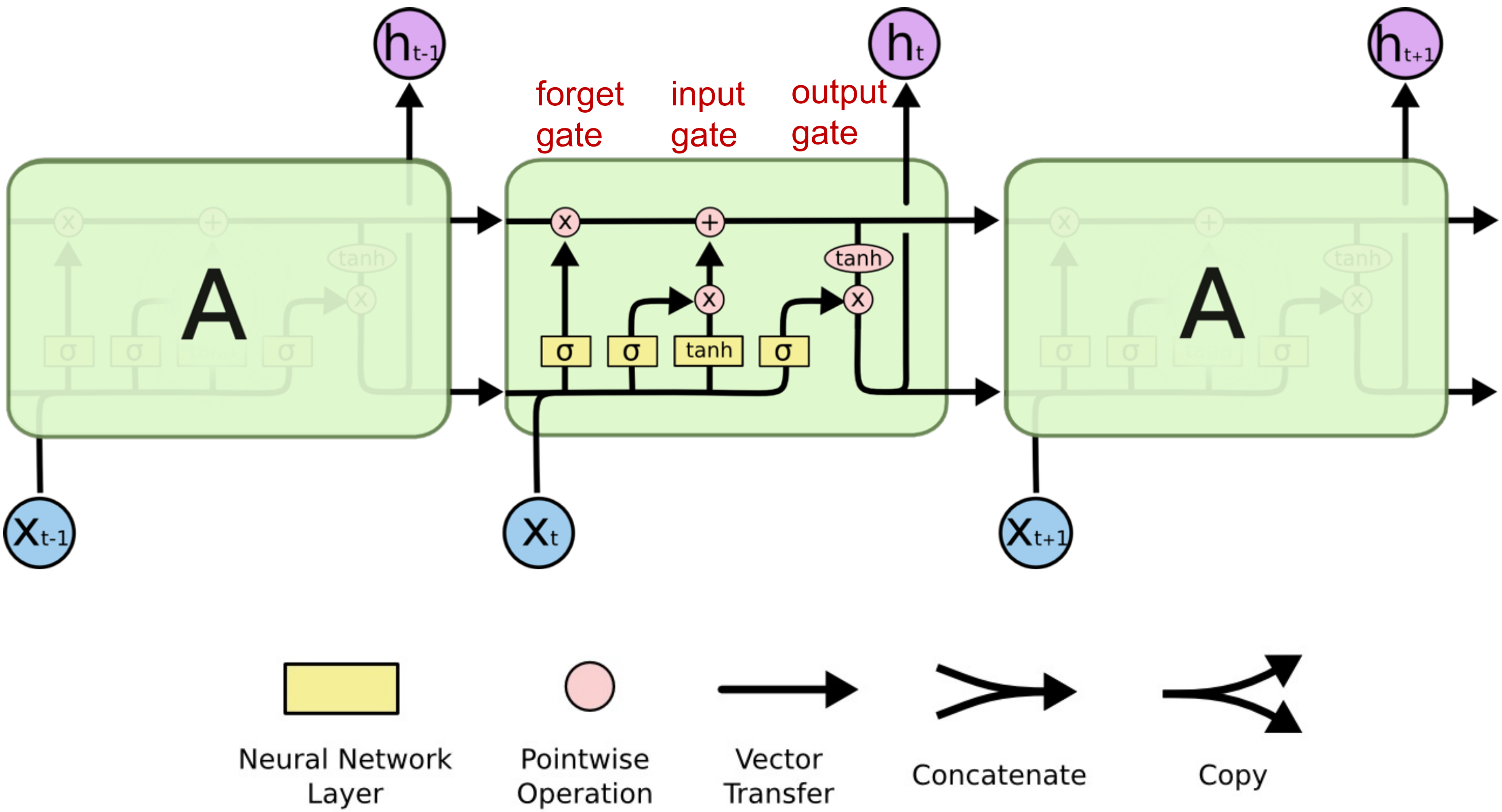
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory (LSTM)



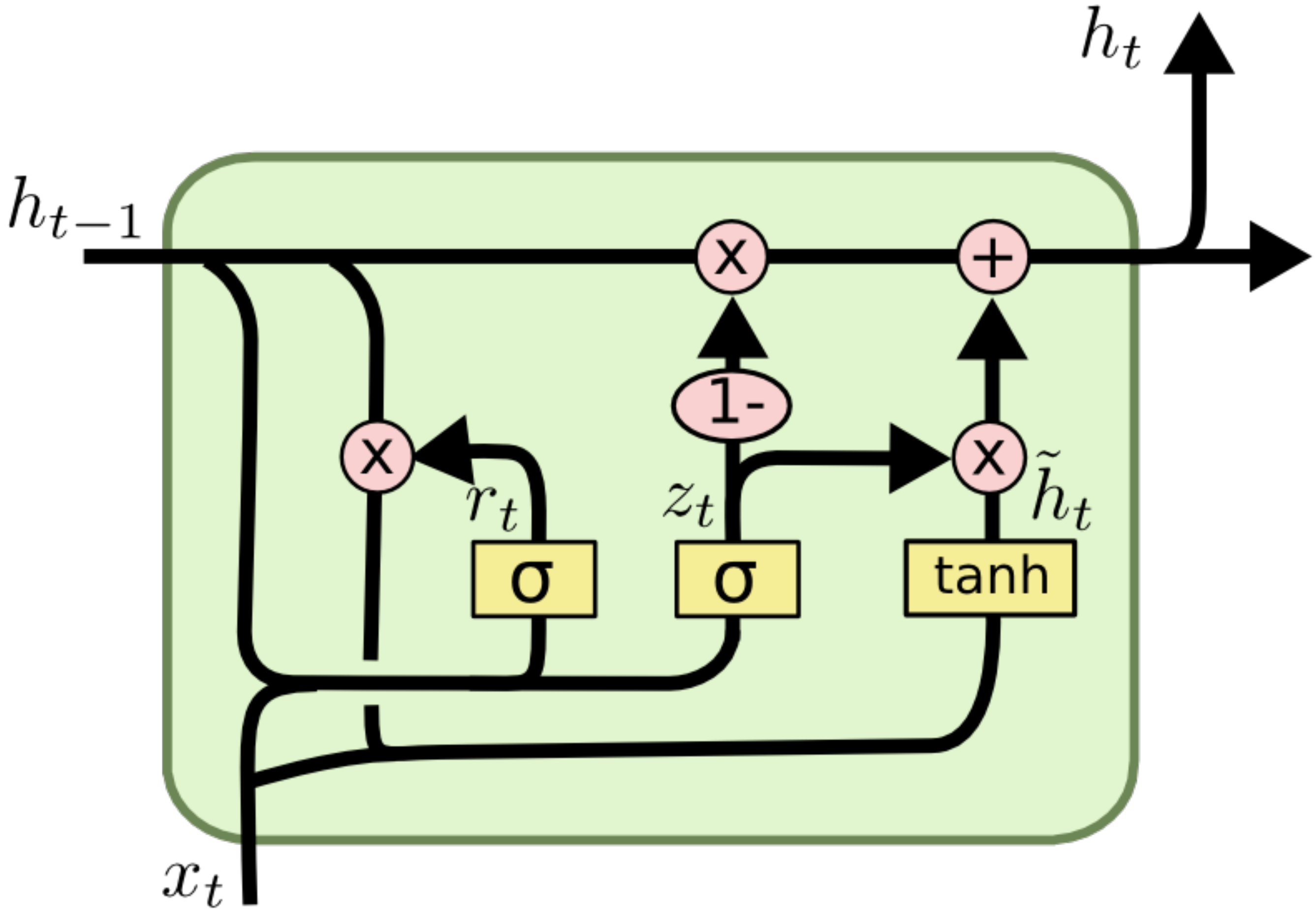
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory (LSTM)



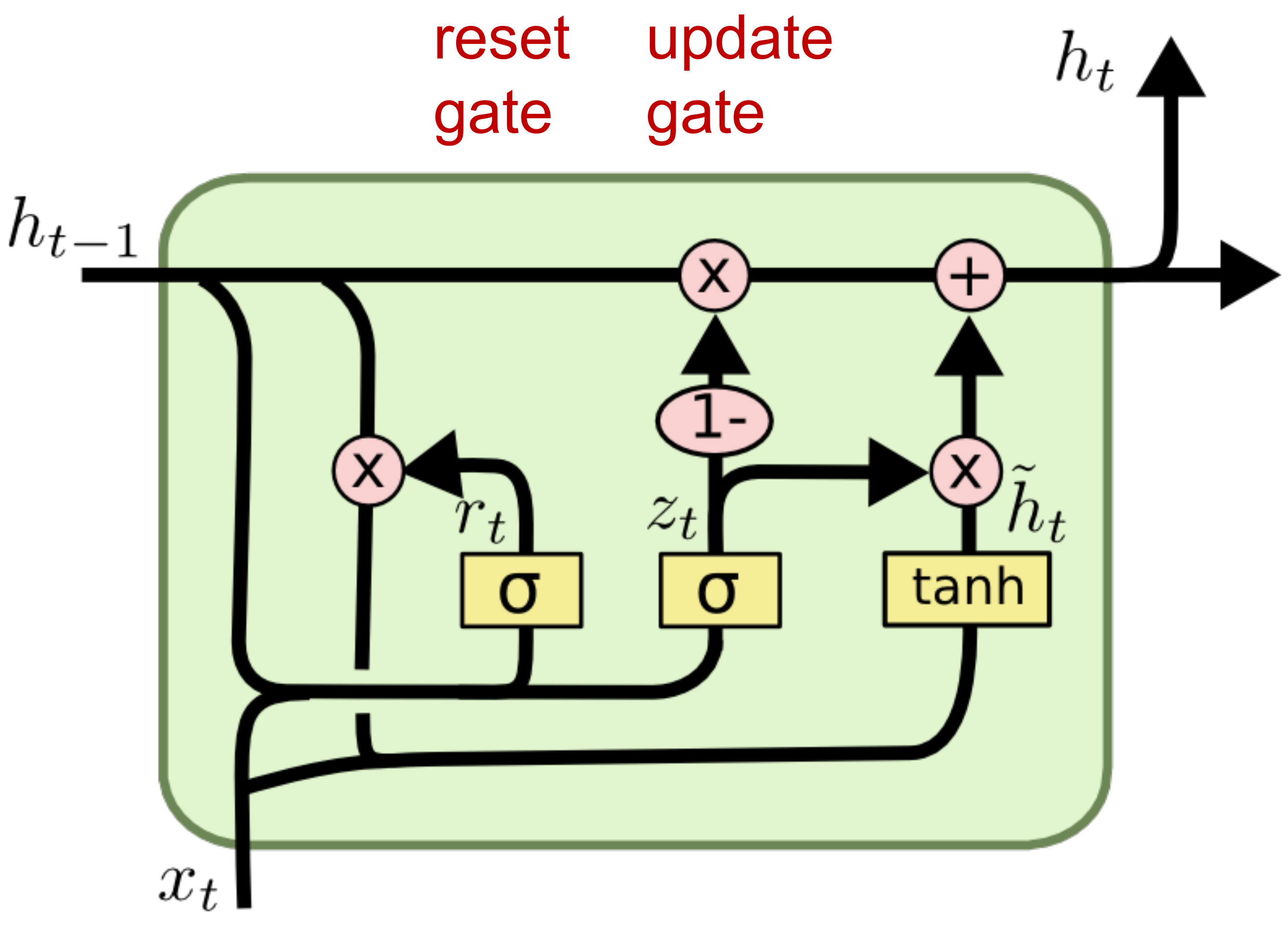
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Gated Recurrent Unit (GRU)



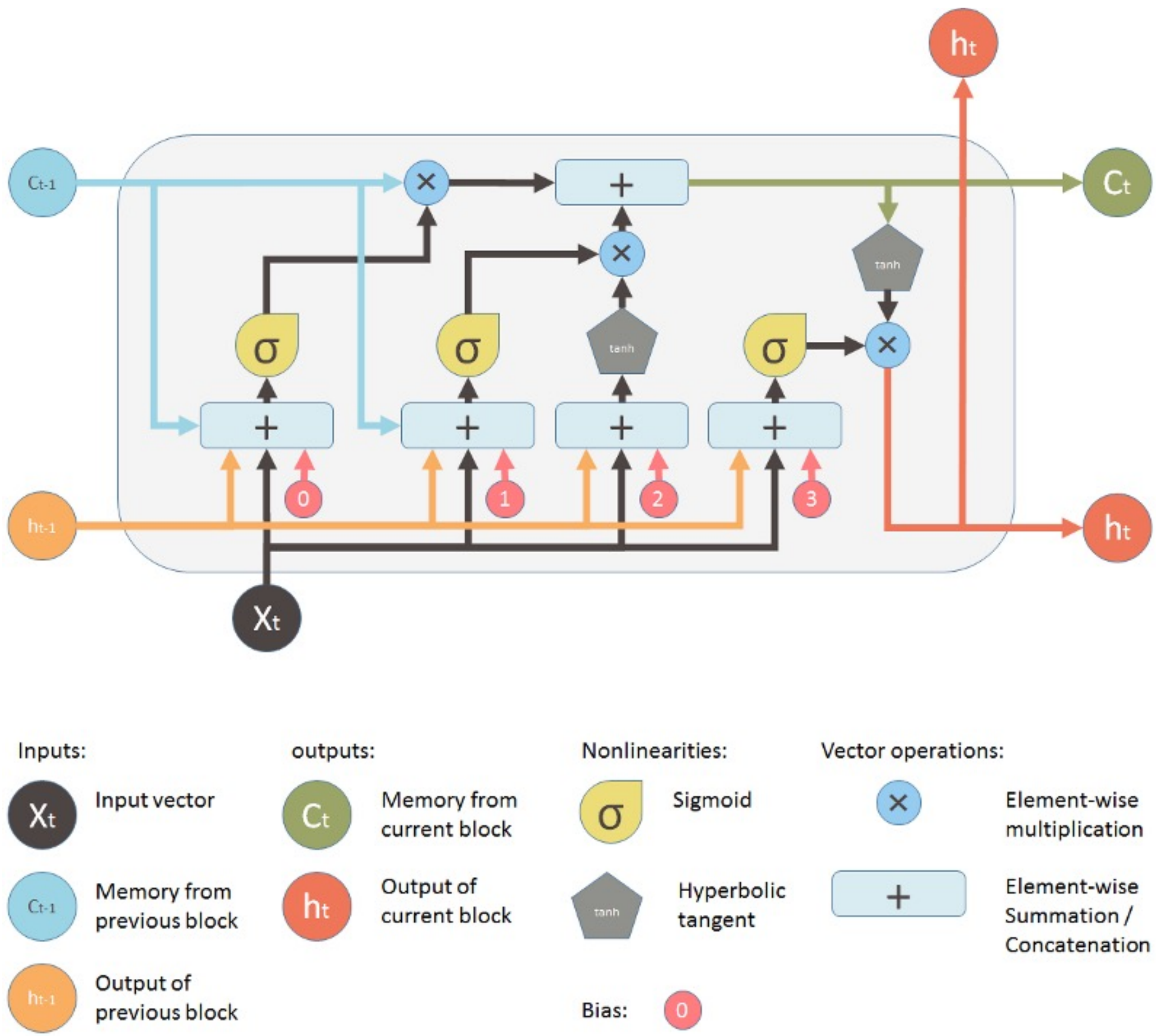
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Gated Recurrent Unit (GRU)



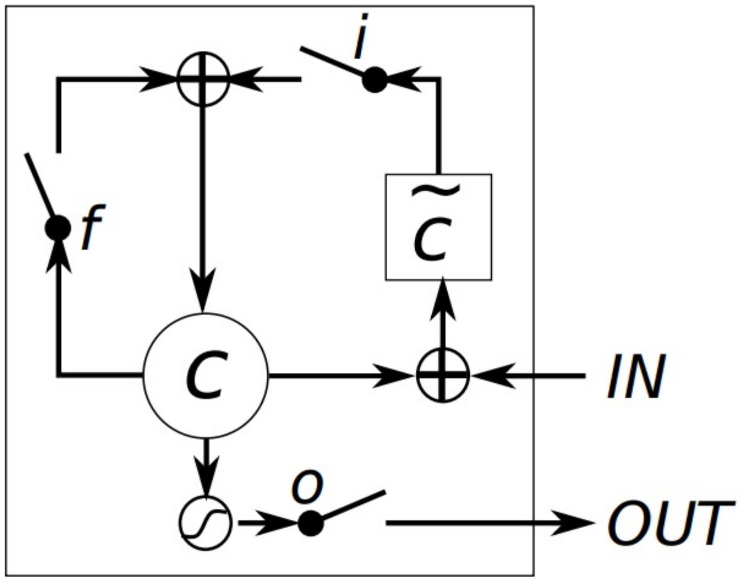
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

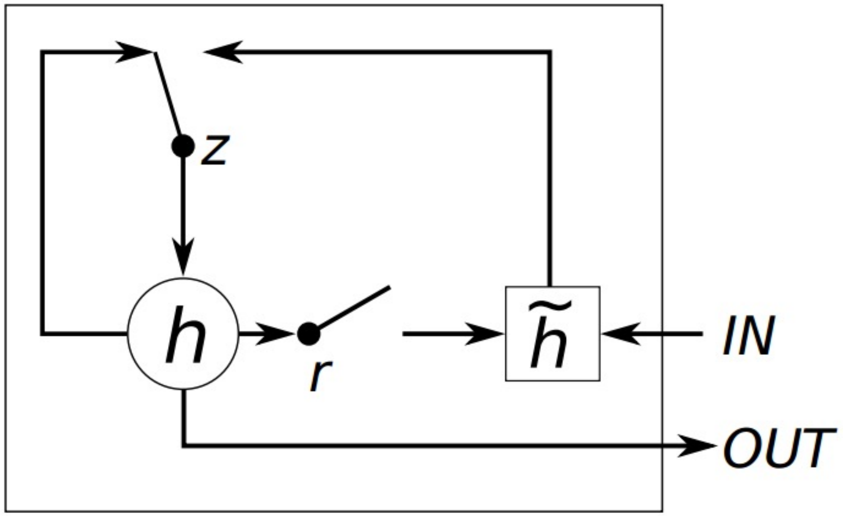
LSTM



Source: Shi Yan (2016), Understanding LSTM and its diagrams, https://medium.com/mlreview/understanding-lstm-and-its-diagrams-37e2f46f1714

LSTM vs GRU





LSTM

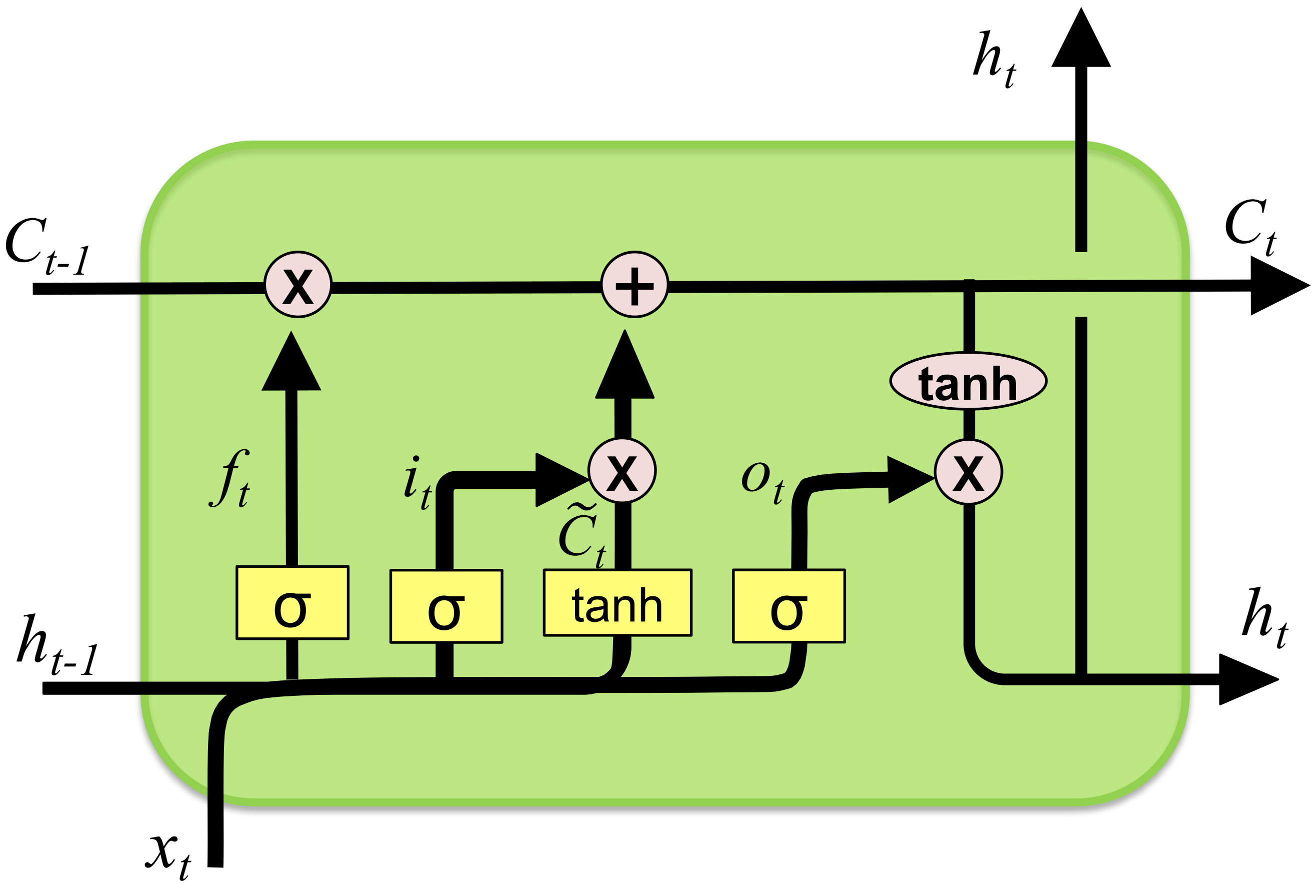
i, f and o are the input, forget and output gates, respectively.
c and c˜ denote the memory cell and the new memory cell content.

GRU

r and z are the reset and update gates, and h and h˜ are the activation and the candidate activation.

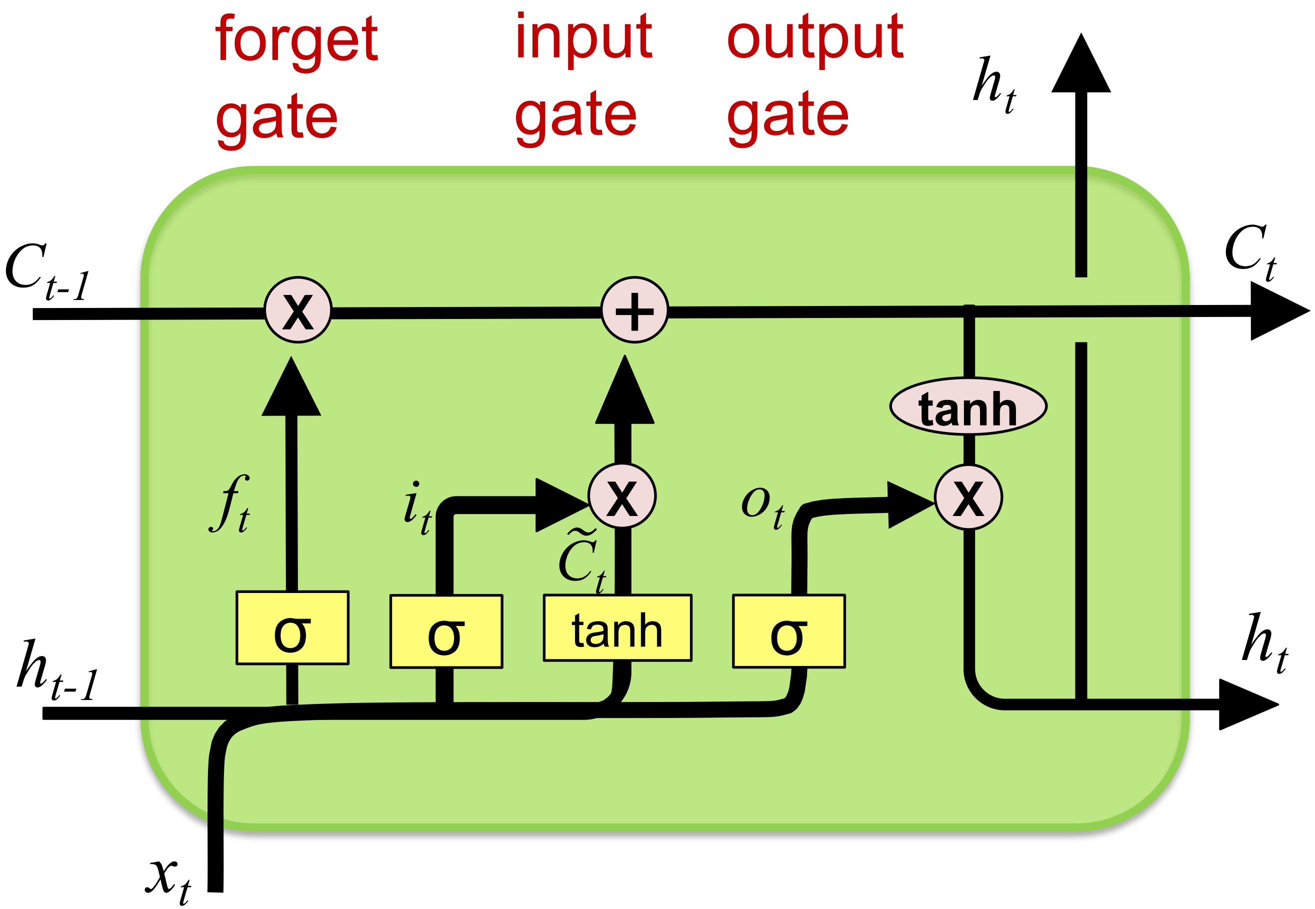
Source: Chung, Junyoung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555* (2014).

Long Short Term Memory (LSTM)



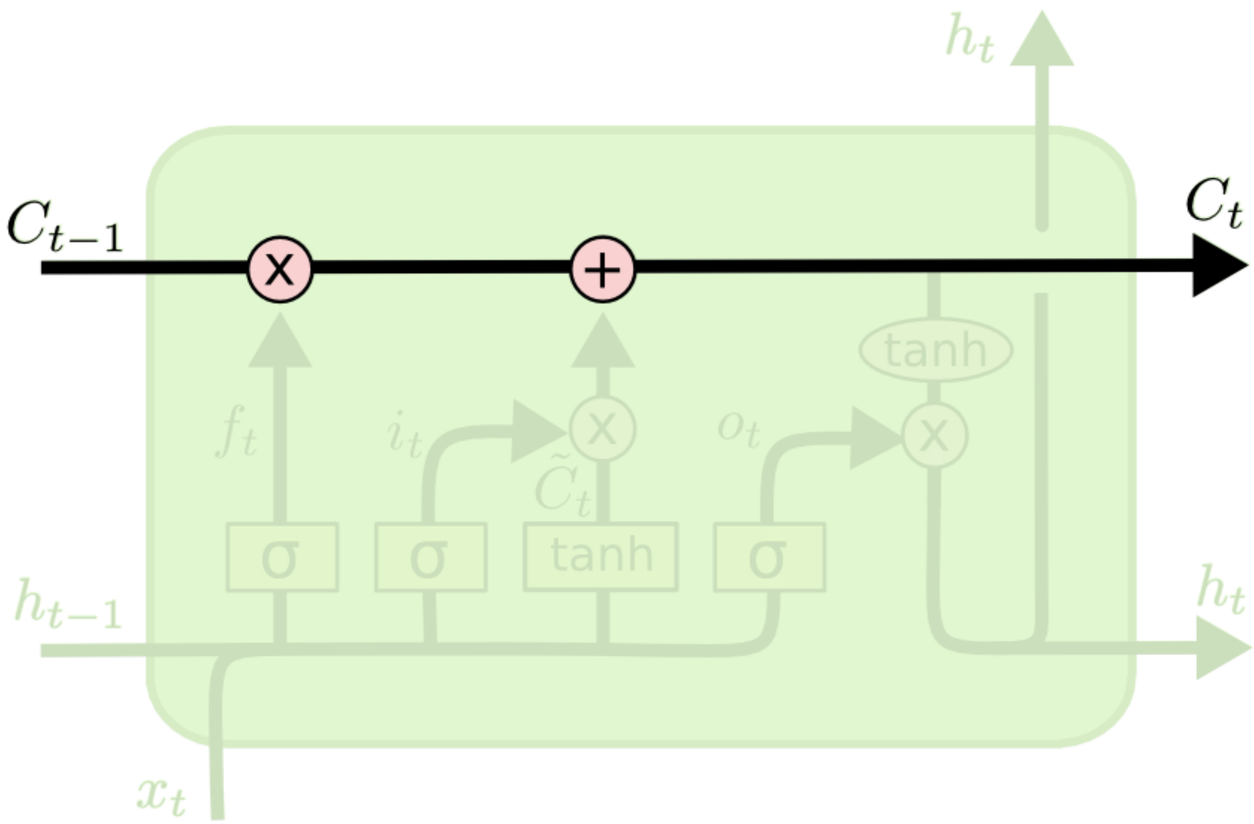
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Long Short Term Memory (LSTM)



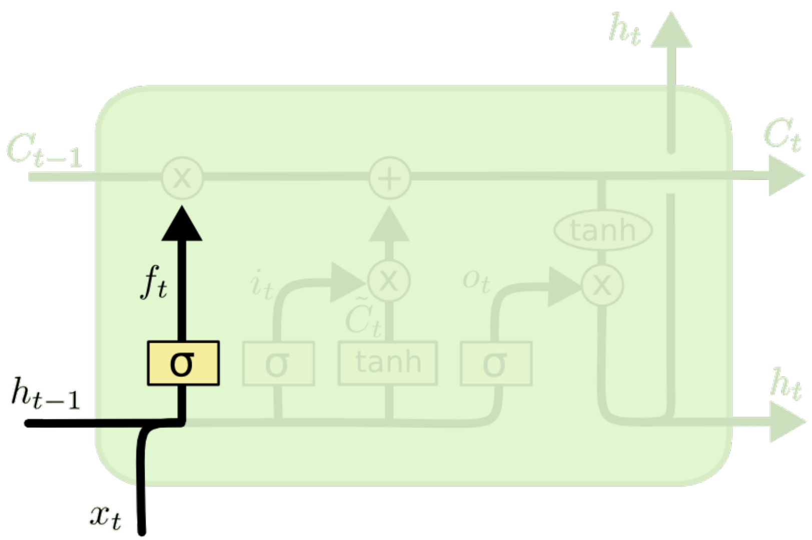
Source: Christopher Olah, (2015) Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTM
Memory state (C)



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

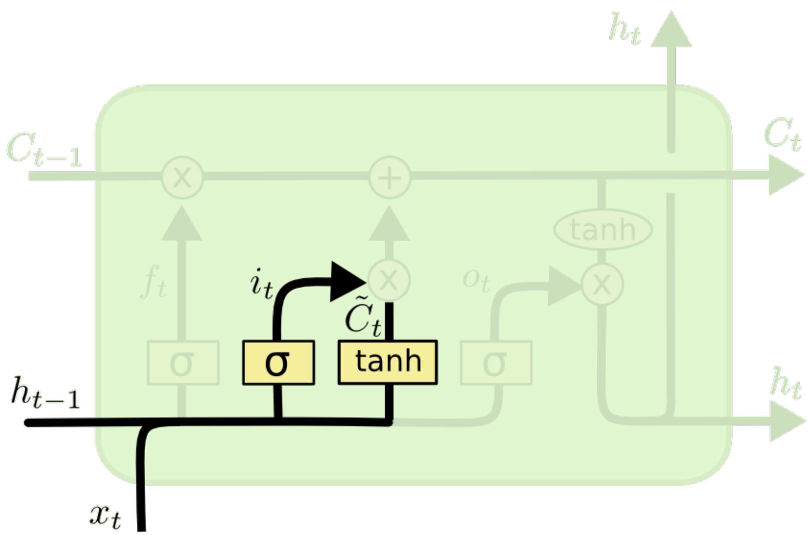
LSTM
forget gate (f)



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

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

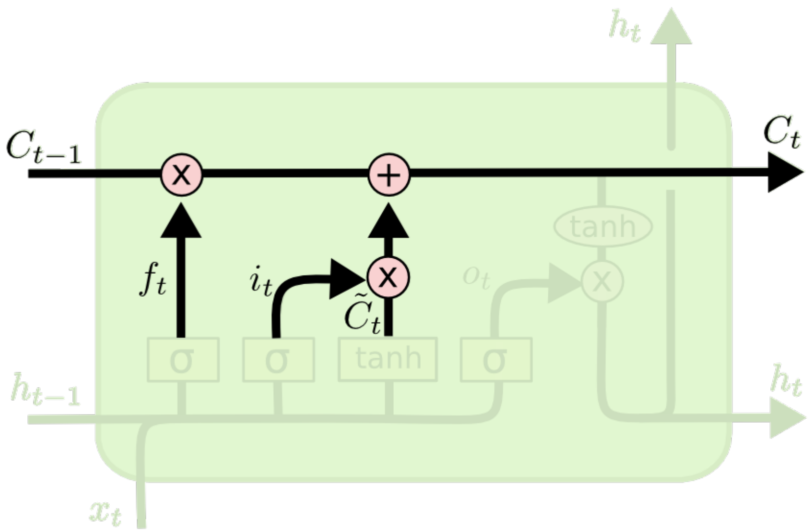
LSTM
input gate (i)



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

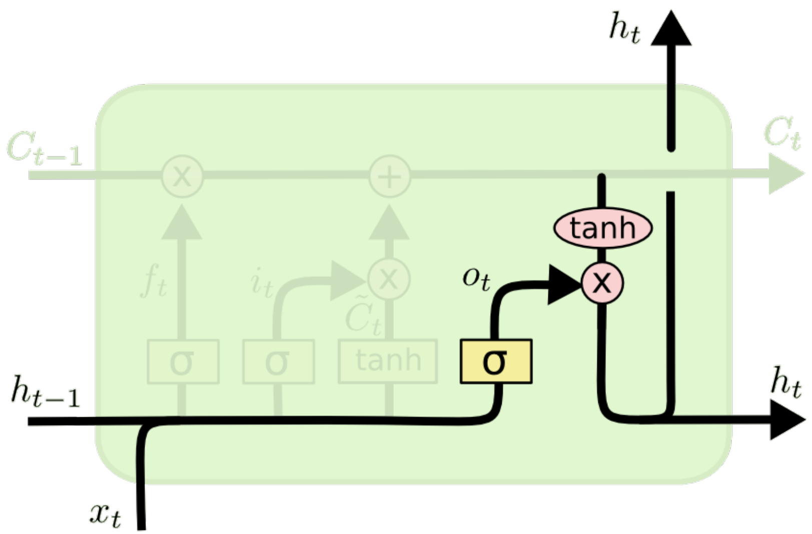
LSTM
Memory state (C)



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

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

LSTM output gate (o)

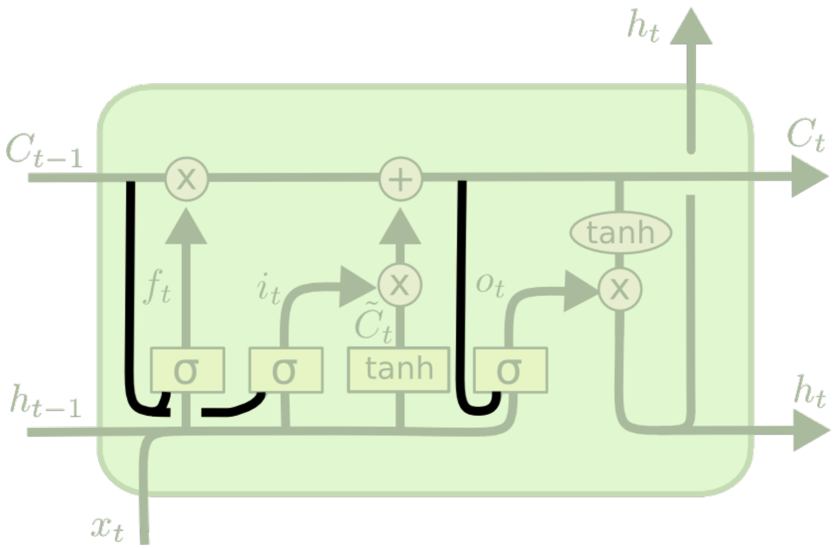


$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

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

LSTM
forget (f), input (i), output (o) gates

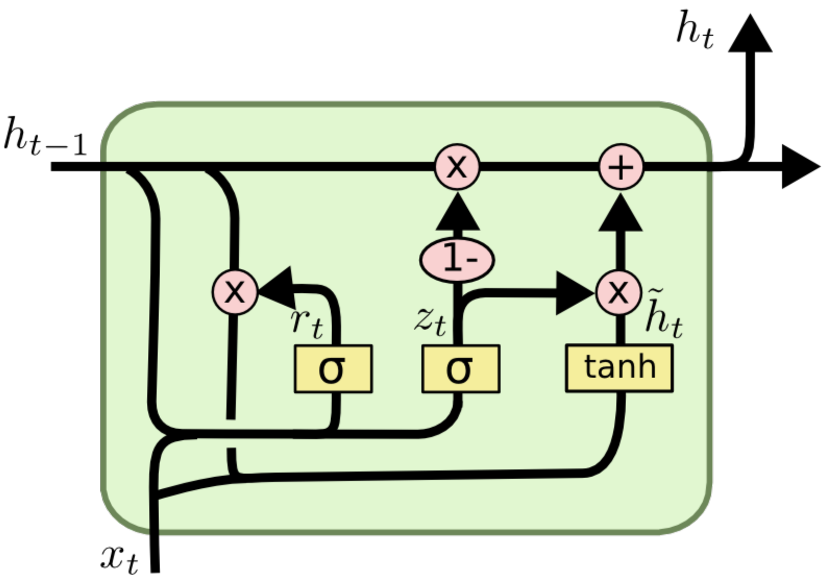


$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \ + \ b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \ + \ b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \ + \ b_o\right)$$

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

Gated Recurrent Unit (GRU)

update (z), reset (r) gates



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

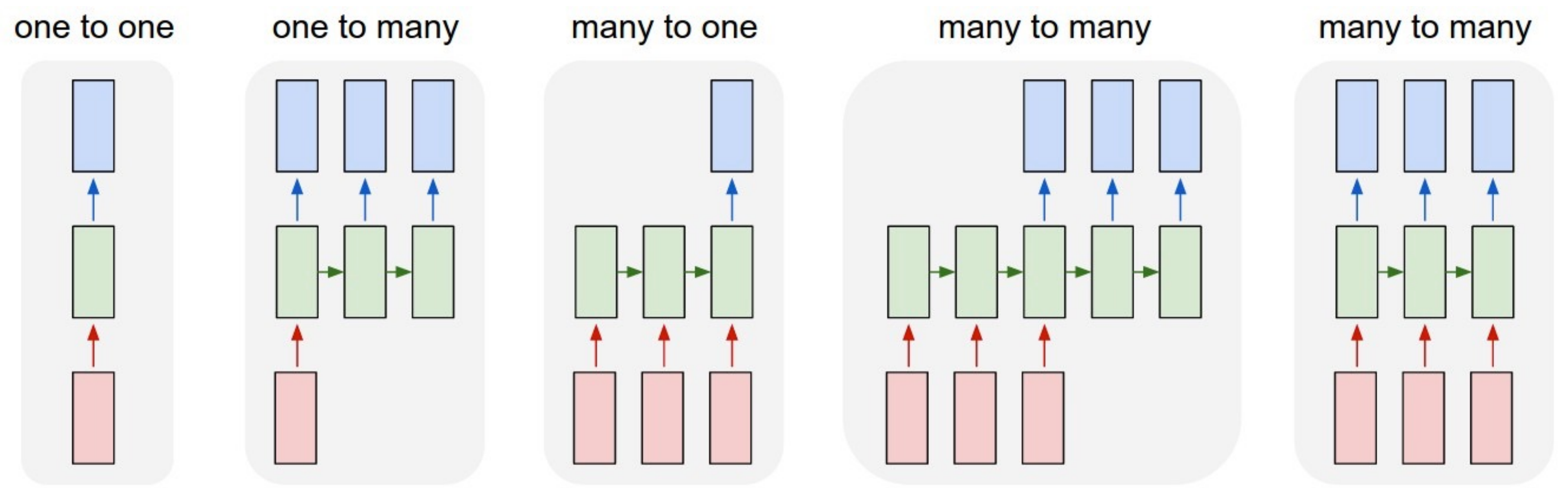
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

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

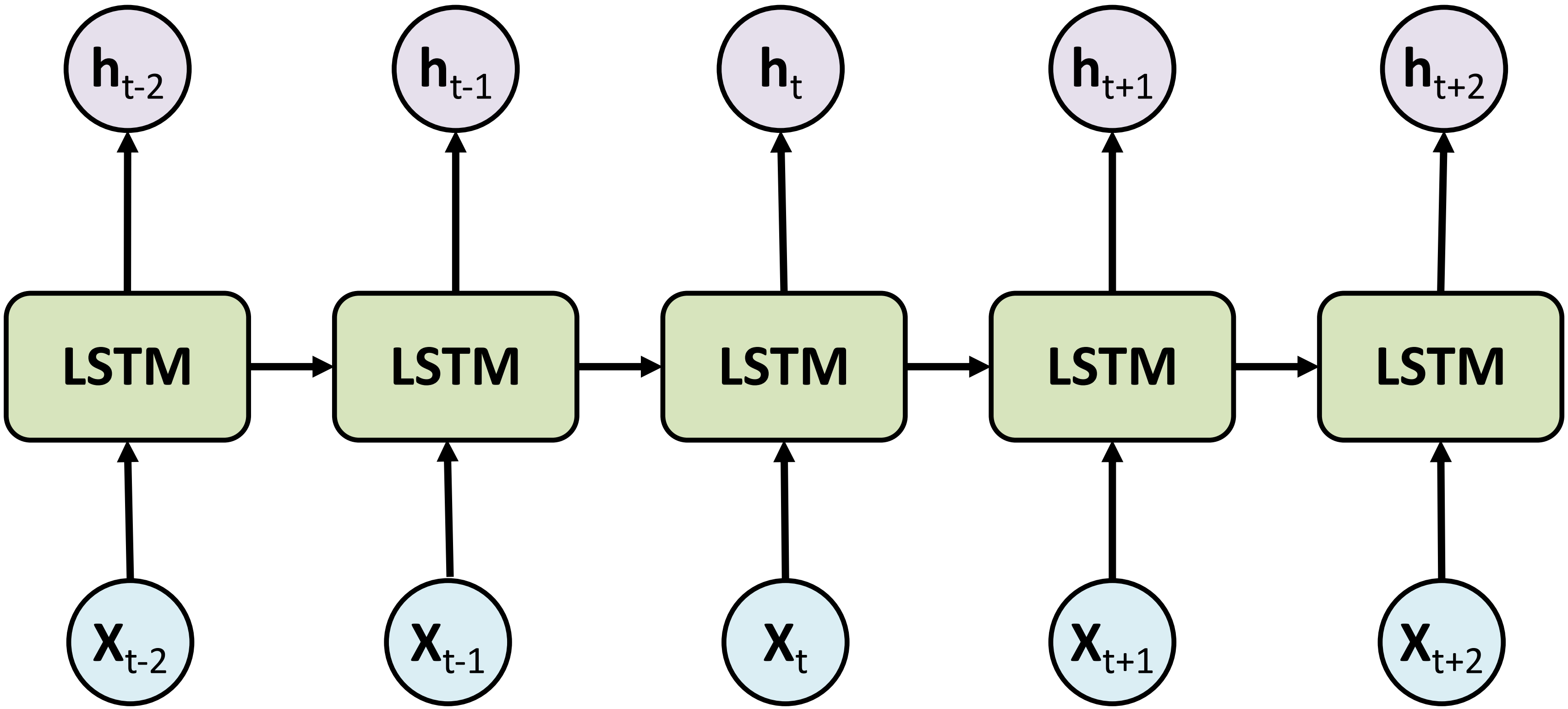
LSTM Recurrent Neural Network



Traditional Neural Network | **Music Generation** | **Sentiment Classification** | **Name Entity Recognition** | **Machine Translation**

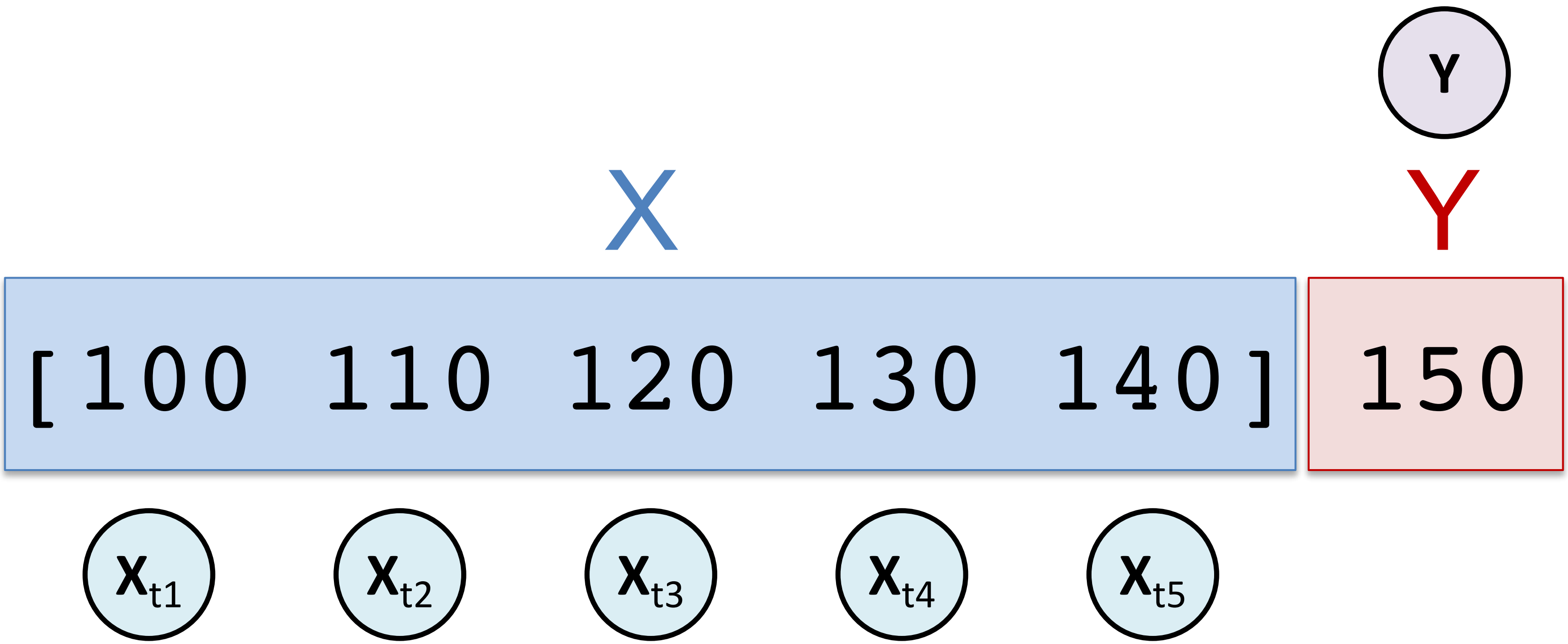
Source: https://github.com/Vict0rSch/deep_learning/tree/master/keras/recurrent

Long Short Term Memory (LSTM) for Time Series Forecasting

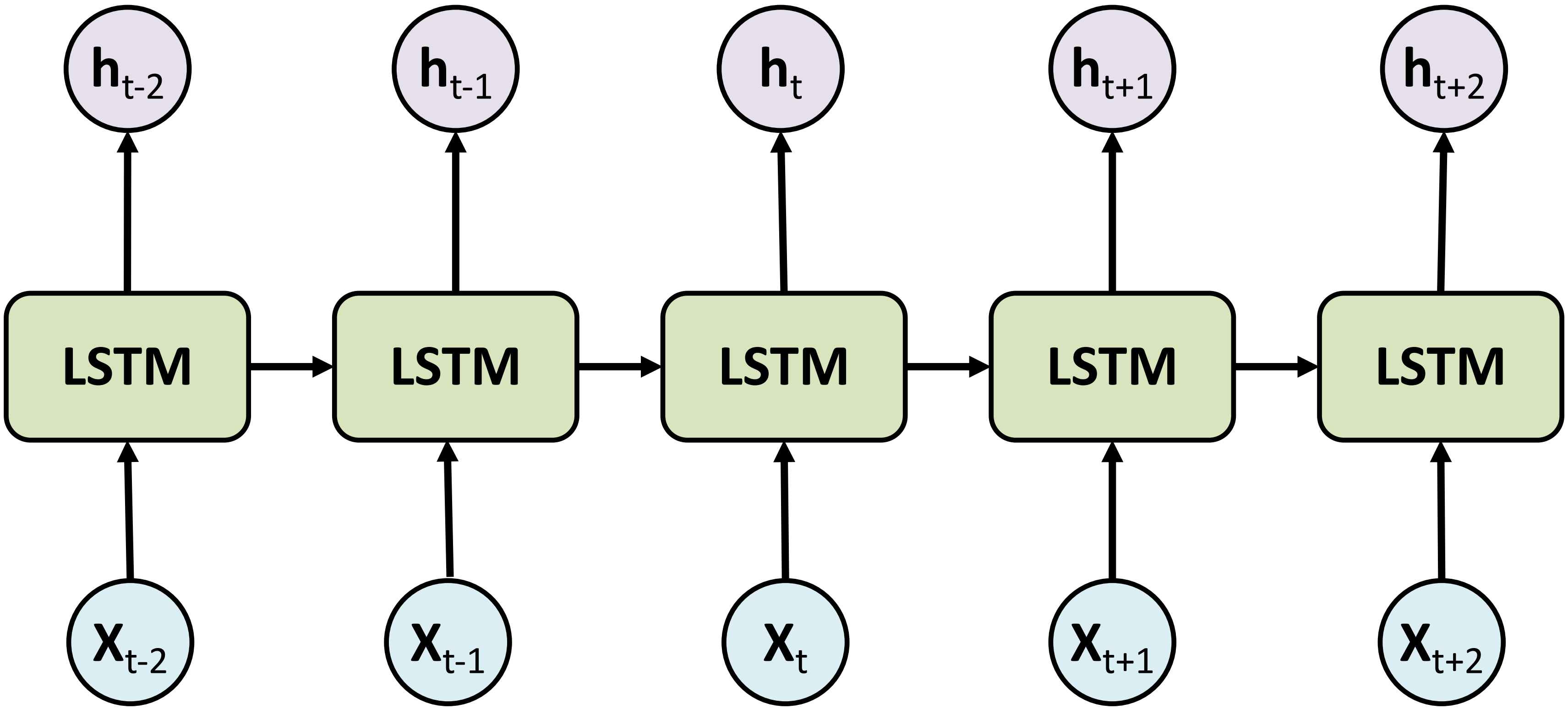


Time Series Data

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



Long Short Term Memory (LSTM) for Time Series Forecasting

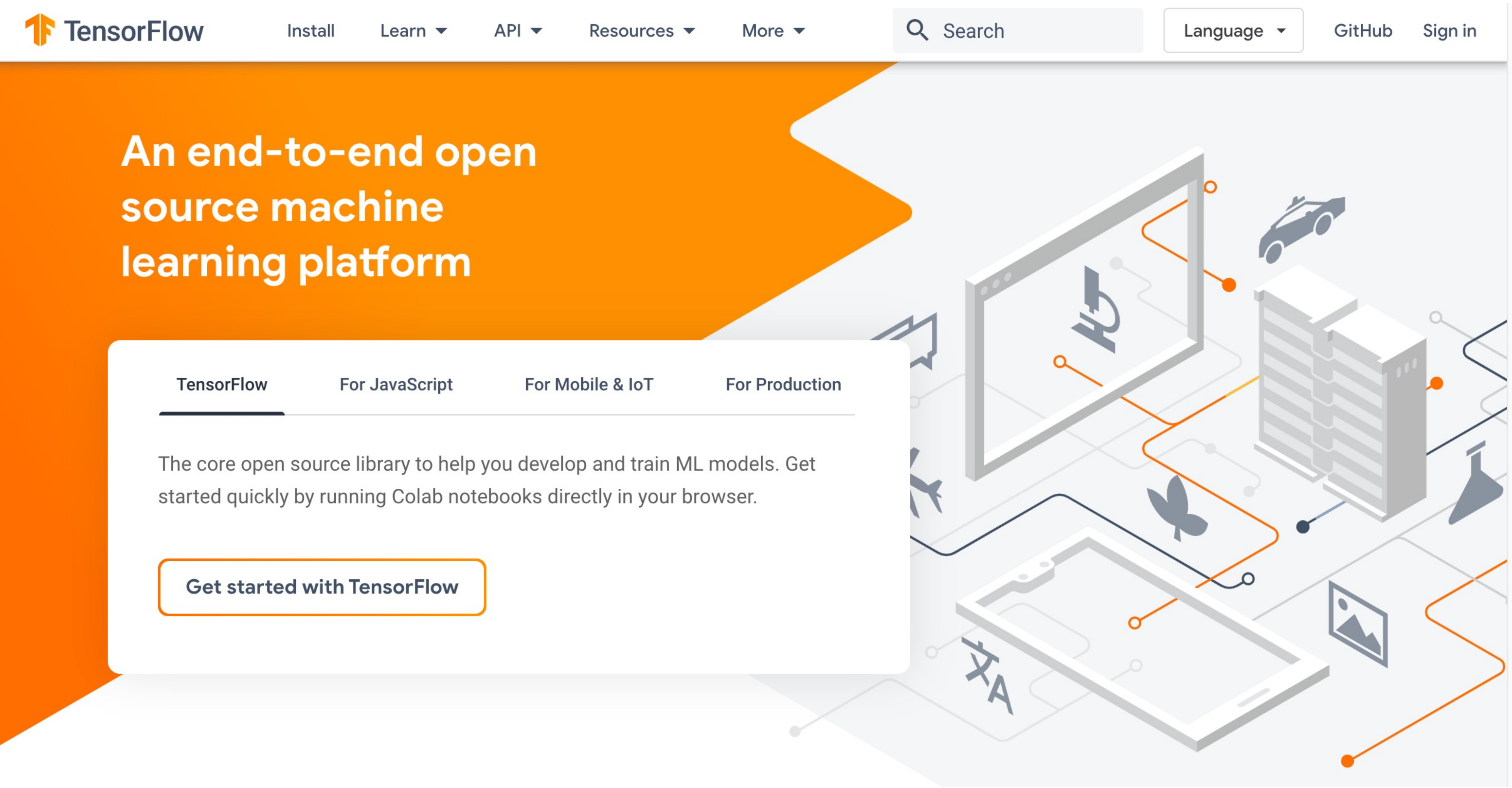


Time Series Data

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

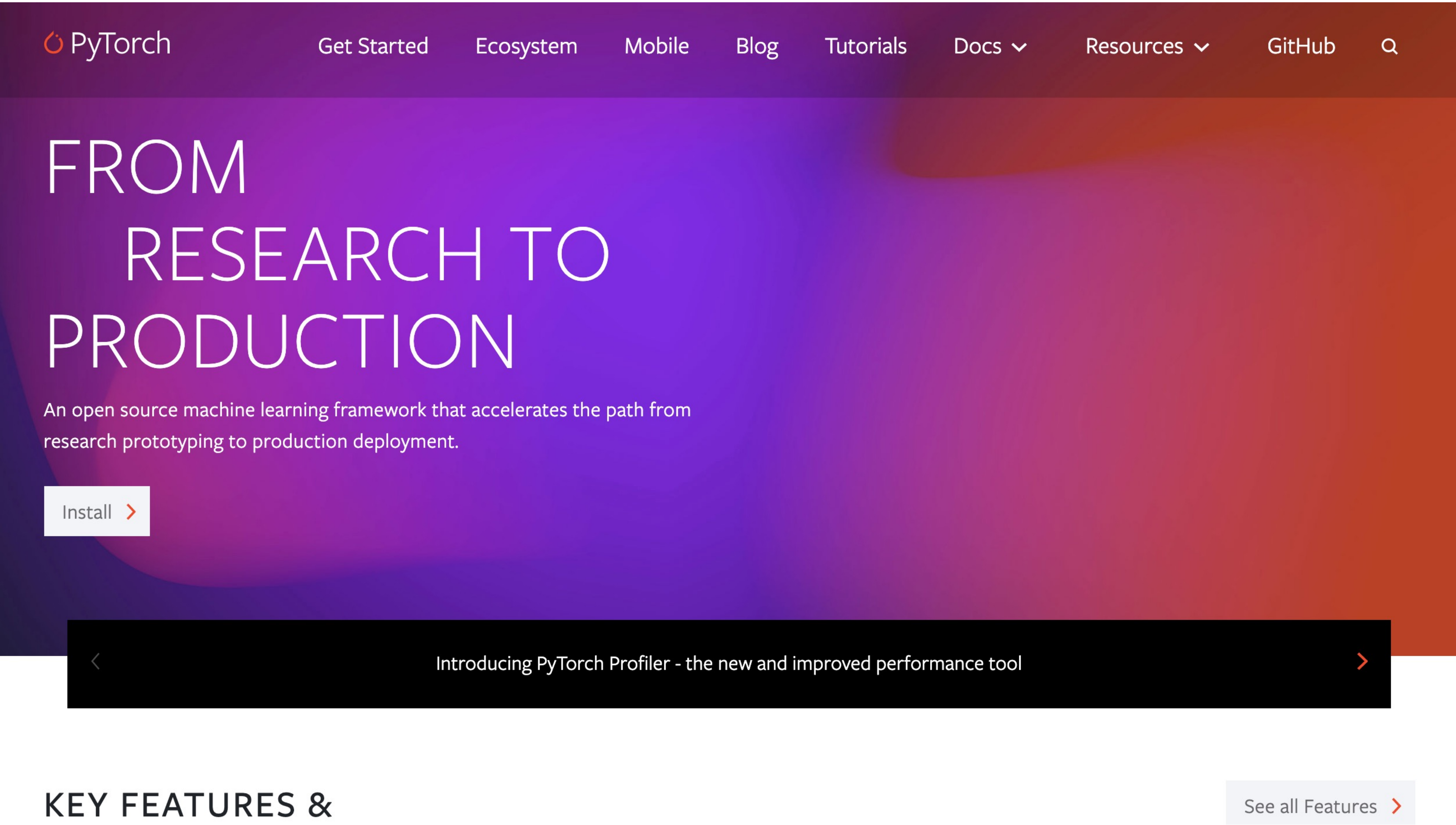
X	Y
[10 20 30]	40
[20 30 40]	50
[30 40 50]	60
[40 50 60]	70
[50 60 70]	80
[60 70 80]	90

TensorFlow



https://www.tensorflow.org/

PyTorch



https://pytorch.org/

TensorFlow

TensorFlow

- An end-to-end open source machine learning platform.
- The core open source library to help you develop and train ML models.
- Get started quickly by running Colab notebooks directly in your browser.

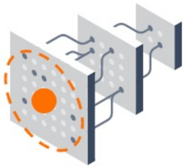
https://www.tensorflow.org/

Why TensorFlow 2.0

Why TensorFlow

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

About →



Easy model building

Build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging.

Robust ML production anywhere

Easily train and deploy models in the cloud, on-prem, in the browser, or on-device no matter what language you use.

Powerful experimentation for research

A simple and flexible architecture to take new ideas from concept to code, to state-of-the-art models, and to publication faster.

TensorFlow

TensorFlow 2.0 vs. 1.X

```
# TensorFlow 2.0
outputs = f(input)

# TensorFlow 1.X
outputs = session.run(f(placeholder), feed_dict={placeholder: input})
```

Source: https://www.tensorflow.org/guide/effective_tf2

TensorFlow

TensorFlow 2.0

```python
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
  tf.keras.layers.Flatten(input_shape=(28, 28)),
  tf.keras.layers.Dense(128, activation='relu'),
  tf.keras.layers.Dropout(0.2),
  tf.keras.layers.Dense(10, activation='softmax')
])

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

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

https://www.tensorflow.org/overview/

TensorFlow

TensorFlow 2 Quick Start

beginner.ipynb

File Edit View Insert Runtime Tools Help Last edited on October 2 by lamberta

Share

A

+ Code + Text Copy to Drive Connect Editing

TensorFlow 2 quickstart for beginners

View on TensorFlow.org Run in Google Colab View source on GitHub Download notebook

This short introduction uses Keras to:

1. Build a neural network that classifies images.
2. Train this neural network.
3. And, finally, evaluate the accuracy of the model.

This is a Google Colaboratory notebook file. Python programs are run directly in the browser—a great way to learn and use TensorFlow. To follow this tutorial, run the notebook in Google Colab by clicking the button at the top of this page.

1. In Colab, connect to a Python runtime: At the top-right of the menu bar, select *CONNECT*.
2. Run all the notebook code cells: Select *Runtime* > *Run all*.

Download and install the TensorFlow 2 package. Import TensorFlow into your program:

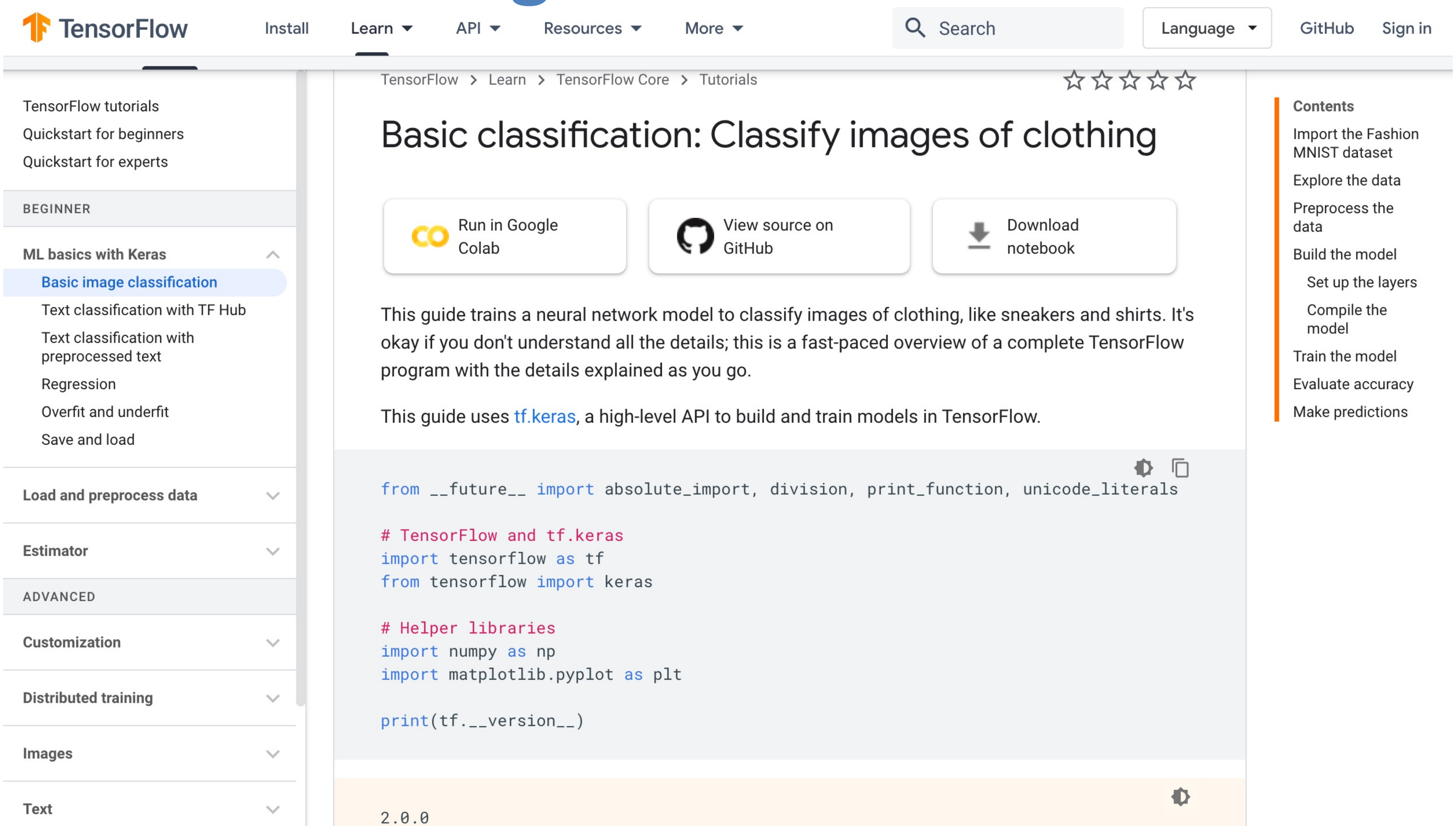
```
from __future__ import absolute_import, division, print_function, unicode_literals

# Install TensorFlow
try:
  # %tensorflow_version only exists in Colab.
  %tensorflow_version 2.x
except Exception:
  pass
```

https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/quickstart/beginner.ipynb

TensorFlow

TensorFlow Image Classification


TensorFlow Install Learn API Resources More Search Language GitHub Sign in

TensorFlow tutorials
Quickstart for beginners
Quickstart for experts

BEGINNER

ML basics with Keras

Basic image classification
Text classification with TF Hub
Text classification with preprocessed text
Regression
Overfit and underfit
Save and load

Load and preprocess data

Estimator

ADVANCED

Customization

Distributed training

Images

Text

TensorFlow > Learn > TensorFlow Core > Tutorials

Basic classification: Classify images of clothing

Run in Google Colab | View source on GitHub | Download notebook

This guide trains a neural network model to classify images of clothing, like sneakers and shirts. It's okay if you don't understand all the details; this is a fast-paced overview of a complete TensorFlow program with the details explained as you go.

This guide uses tf.keras, a high-level API to build and train models in TensorFlow.

```
from __future__ import absolute_import, division, print_function, unicode_literals

# TensorFlow and tf.keras
import tensorflow as tf
from tensorflow import keras

# Helper libraries
import numpy as np
import matplotlib.pyplot as plt

print(tf.__version__)
```

```
2.0.0
```

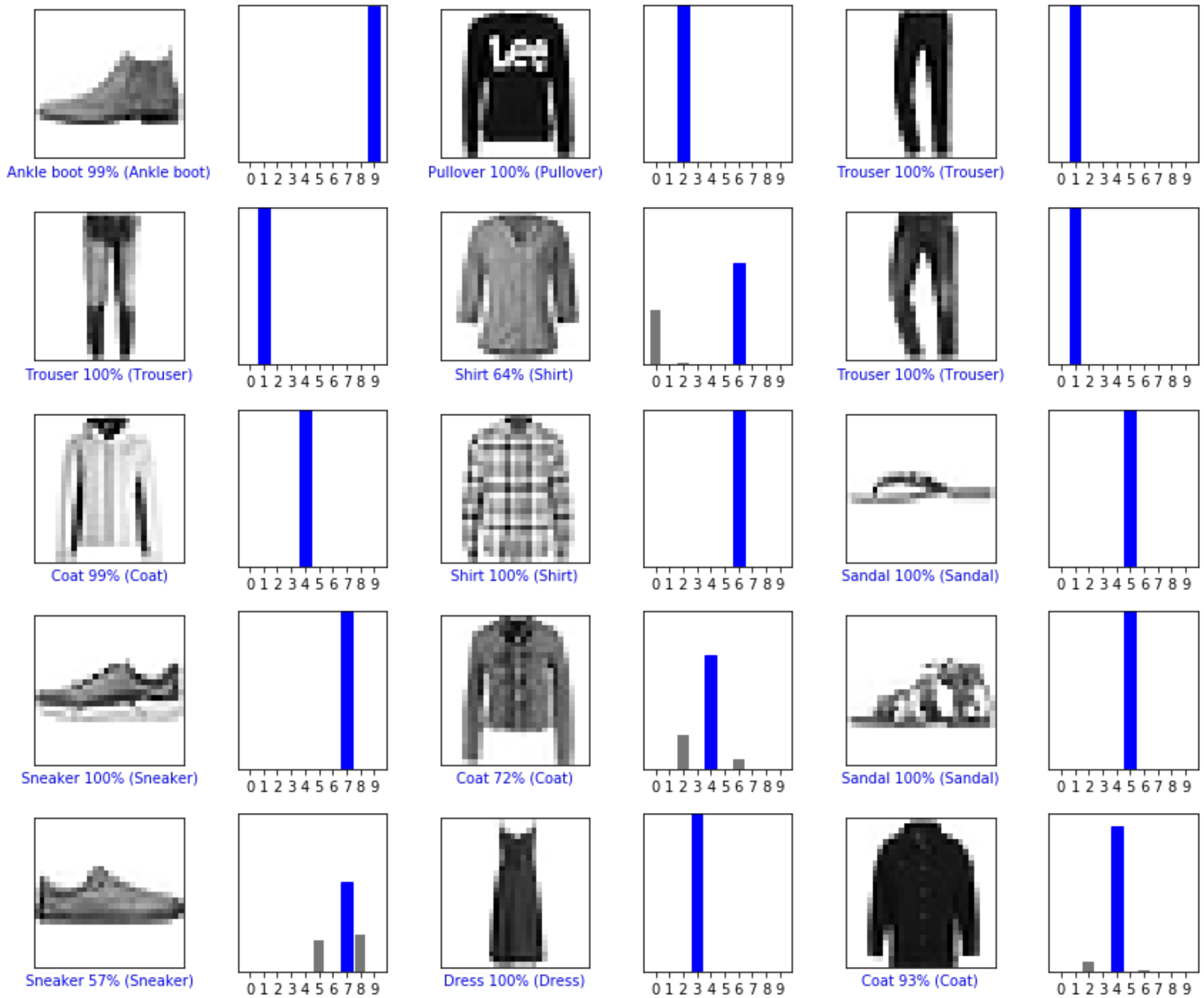
Contents

- Import the Fashion MNIST dataset
- Explore the data
- Preprocess the data
- Build the model
  - Set up the layers
  - Compile the model
- Train the model
- Evaluate accuracy
- Make predictions

https://www.tensorflow.org/tutorials/keras/classification

TensorFlow

Image Classification Fashion MNIST dataset



https://www.tensorflow.org/tutorials/keras/classification

TensorFlow

Text Classification with TF Hub

TensorFlow Install Learn API Resources More Search Language GitHub Sign in

Overview Tutorials Guide TF 1

TensorFlow tutorials
Quickstart for beginners
Quickstart for experts

BEGINNER

ML basics with Keras
- Basic image classification
- Text classification with TF Hub
- Text classification with preprocessed text
- Regression
- Overfit and underfit
- Save and load

Load and preprocess data
- CSV
- NumPy
- pandas.DataFrame
- Images
- Text
- Unicode
- TF.Text
- TFRecord and tf.Example
- Additional formats with tf.io

TensorFlow > Learn > TensorFlow Core > Tutorials

Text classification with TensorFlow Hub: Movie reviews

Run in Google Colab | View source on GitHub | Download notebook

This notebook classifies movie reviews as *positive* or *negative* using the text of the review. This is an example of *binary*—or two-class—classification, an important and widely applicable kind of machine learning problem.

The tutorial demonstrates the basic application of transfer learning with TensorFlow Hub and Keras.

We'll use the IMDB dataset that contains the text of 50,000 movie reviews from the Internet Movie Database. These are split into 25,000 reviews for training and 25,000 reviews for testing. The training and testing sets are *balanced*, meaning they contain an equal number of positive and negative reviews.

This notebook uses tf.keras, a high-level API to build and train models in TensorFlow, and TensorFlow Hub, a library and platform for transfer learning. For a more advanced text classification tutorial using `tf.keras`, see the MLCC Text Classification Guide.

```
from __future__ import absolute_import, division, print_function, unicode_literals
```

Contents
- Download the IMDB dataset
- Explore the data
- Build the model
  - Loss function and optimizer
- Train the model
- Evaluate the model
- Further reading

https://www.tensorflow.org/tutorials/keras/text_classification_with_hub

TensorFlow

Text Classification with Pre Text

TensorFlow Install Learn API Resources More Search Language GitHub Sign in

TensorFlow tutorials
Quickstart for beginners
Quickstart for experts

BEGINNER

ML basics with Keras
- Basic image classification
- Text classification with TF Hub
- Text classification with preprocessed text
- Regression
- Overfit and underfit
- Save and load

Load and preprocess data
- CSV
- NumPy
- pandas.DataFrame
- Images
- Text
- Unicode
- TF.Text
- TFRecord and tf.Example
- Additional formats with tf.io

Estimator

TensorFlow > Learn > TensorFlow Core > Tutorials

Text classification with preprocessed text: Movie reviews

Run in Google Colab | View source on GitHub | Download notebook

This notebook classifies movie reviews as *positive* or *negative* using the text of the review. This is an example of *binary*—or two-class—classification, an important and widely applicable kind of machine learning problem.

We'll use the IMDB dataset that contains the text of 50,000 movie reviews from the Internet Movie Database. These are split into 25,000 reviews for training and 25,000 reviews for testing. The training and testing sets are *balanced*, meaning they contain an equal number of positive and negative reviews.

This notebook uses tf.keras, a high-level API to build and train models in TensorFlow. For a more advanced text classification tutorial using `tf.keras`, see the MLCC Text Classification Guide.

Setup

```
from __future__ import absolute_import, division, print_function, unicode_literals
```

Contents

Setup
Download the IMDB dataset
Try the encoder
Explore the data
Prepare the data for training
Build the model
Hidden units
Loss function and optimizer
Train the model
Evaluate the model
Create a graph of accuracy and loss over time

https://www.tensorflow.org/tutorials/keras/text_classification

TensorFlow

Regression

TensorFlow | Install | Learn | API | Resources | More | Search | Language | GitHub | Sign in

TensorFlow tutorials
Quickstart for beginners
Quickstart for experts

BEGINNER

ML basics with Keras
- Basic image classification
- Text classification with TF Hub
- Text classification with preprocessed text
- Regression
- Overfit and underfit
- Save and load

Load and preprocess data

Estimator

ADVANCED

Customization

Distributed training

Images

Text

TensorFlow > Learn > TensorFlow Core > Tutorials

Basic regression: Predict fuel efficiency

Run in Google Colab | View source on GitHub | Download notebook

In a *regression* problem, we aim to predict the output of a continuous value, like a price or a probability. Contrast this with a *classification* problem, where we aim to select a class from a list of classes (for example, where a picture contains an apple or an orange, recognizing which fruit is in the picture).

This notebook uses the classic Auto MPG Dataset and builds a model to predict the fuel efficiency of late-1970s and early 1980s automobiles. To do this, we'll provide the model with a description of many automobiles from that time period. This description includes attributes like: cylinders, displacement, horsepower, and weight.

This example uses the `tf.keras` API, see this guide for details.

```
# Use seaborn for pairplot
!pip install -q seaborn
```

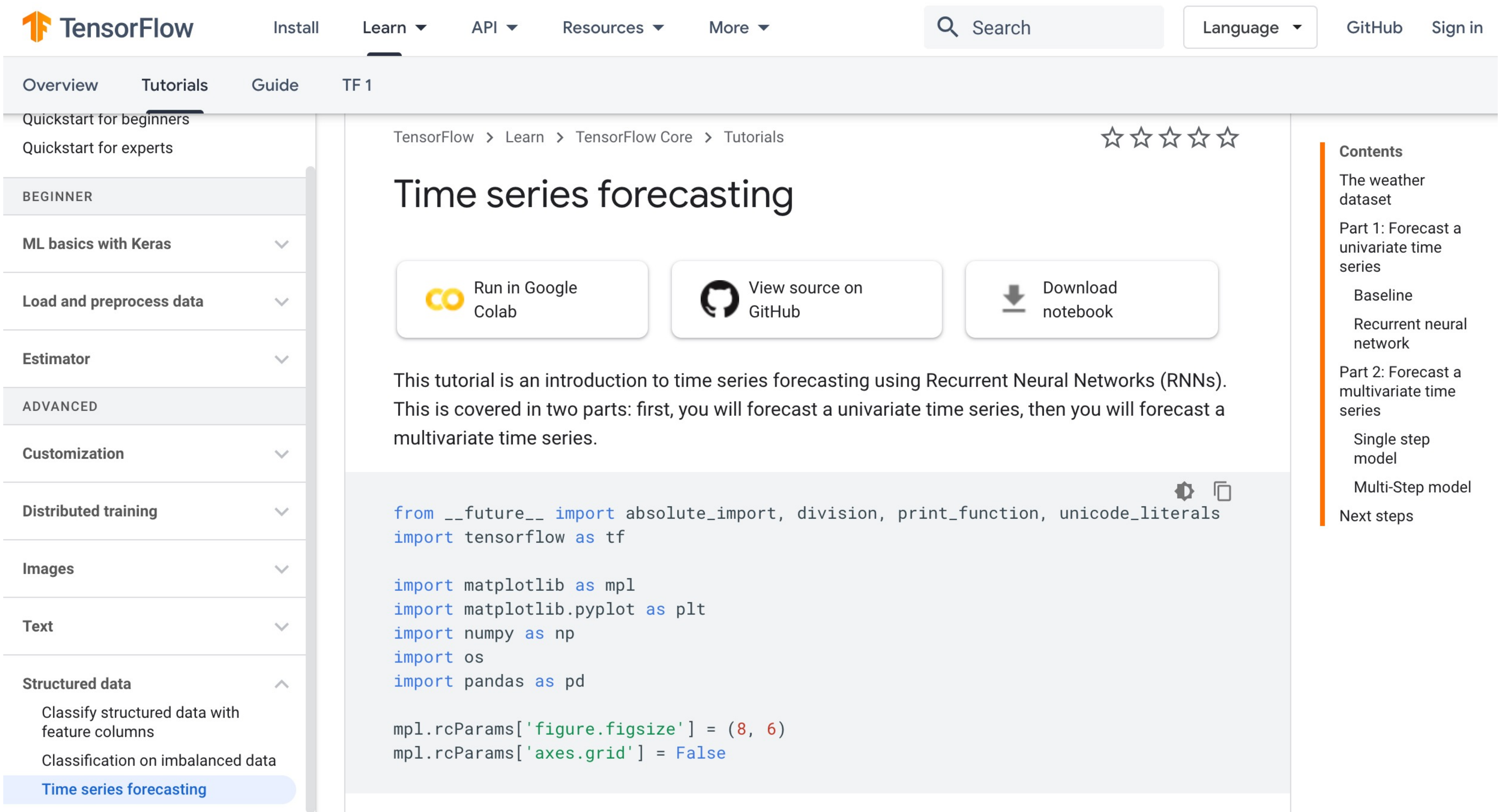
```
from __future__ import absolute_import, division, print_function, unicode_literals

import pathlib
```


Contents
- The Auto MPG dataset
  - Get the data
  - Clean the data
  - Split the data into train and test
  - Inspect the data
  - Split features from labels
  - Normalize the data
- The model
  - Build the model
  - Inspect the model
  - Train the model
  - Make predictions
- Conclusion

https://www.tensorflow.org/tutorials/keras/regression

TensorFlow 2.0 Time Series Forecasting



https://www.tensorflow.org/tutorials/structured_data/time_series

Basic Classification
Fashion MNIST Image Classification

https://colab.research.google.com/drive/19PJOJi1vn1kjcutIzNHjRSLbeVl4kd5z

CO tf01_basic_classification.ipynb ☆

File Edit View Insert Runtime Tools Help

COMMENT SHARE A

CODE TEXT CELL CELL CONNECT EDITING

Table of contents Code snippets Files

Copyright 2018 The TensorFlow Authors.

Licensed under the Apache License, Version 2.0 (the "License");

MIT License

Train your first neural network: basic classification

Import the Fashion MNIST dataset

Explore the data

Preprocess the data

Build the model

Setup the layers

Compile the model

Train the model

Evaluate accuracy

Make predictions

SECTION

Copyright 2018 The TensorFlow Authors.

↳ 2 cells hidden

Train your first neural network: basic classification

View on TensorFlow.org Run in Google Colab View source on GitHub

This guide trains a neural network model to classify images of clothing, like sneakers and shirts. It's okay if you don't understand all the details, this is a fast-paced overview of a complete TensorFlow program with the details explained as we go.

This guide uses tf.keras, a high-level API to build and train models in TensorFlow.

```
# memory footprint support libraries/code
!ln -sf /opt/bin/nvidia-smi /usr/bin/nvidia-smi
!pip install gputil
!pip install psutil
!pip install humanize
import psutil
import humanize
import os
import GPUtil as GPU
GPUs = GPU.getGPUs()
gpu = GPUs[0]
def printm():
    process = psutil.Process(os.getpid())
    print("Gen RAM Free: " + humanize.naturalsize( psutil.virtual_memory().available ), " | Pro
    print("GPU RAM Free: {0:.0f}MB | Used: {1:.0f}MB | Util {2:3.0f}% | Total {3:.0f}MB".format
printm()
```

Source: https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/keras/basic_classification.ipynb

Text Classification
IMDB Movie Reviews

https://colab.research.google.com/drive/1x16h1GhHsLIrLYtPCvCHaoO1W-i_gror

CO tf02_basic-text-classification.ipynb ☆

File Edit View Insert Runtime Tools Help

COMMENT SHARE A

CODE TEXT CELL CELL CONNECT EDITING

Table of contents Code snippets Files

Copyright 2018 The TensorFlow Authors.

Licensed under the Apache License, Version 2.0 (the "License");

MIT License

Text classification with movie reviews

Download the IMDB dataset

Explore the data

Convert the integers back to words

Prepare the data

Build the model

Hidden units

Loss function and optimizer

Create a validation set

Train the model

Evaluate the model

Copyright 2018 The TensorFlow Authors.

↳ 2 cells hidden

Text classification with movie reviews

View on TensorFlow.org Run in Google Colab View source on GitHub

This notebook classifies movie reviews as *positive* or *negative* using the text of the review. This is an example of *binary*—or two-class—classification, an important and widely applicable kind of machine learning problem.

We'll use the IMDB dataset that contains the text of 50,000 movie reviews from the Internet Movie Database. These are split into 25,000 reviews for training and 25,000 reviews for testing. The training and testing sets are *balanced*, meaning they contain an equal number of positive and negative reviews.

This notebook uses tf.keras, a high-level API to build and train models in TensorFlow. For a more advanced text classification tutorial using `tf.keras`, see the MLCC Text Classification Guide.

```
# memory footprint support libraries/code
!ln -sf /opt/bin/nvidia-smi /usr/bin/nvidia-smi
!pip install gputil
!pip install psutil
!pip install humanize
import psutil
import humanize
import os
import GPUtil as GPU
GPUs = GPU.getGPUs()
gpu = GPUs[0]
def printm():
    process = psutil.Process(os.getpid())
```

Source: https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/keras/basic_text_classification.ipynb

Basic Regression
Predict House Prices

https://colab.research.google.com/drive/1v4c8ZHTnRtgd2_25K_AURjR6SCVBRdlj

CO tf03_basic-regression.ipynb ☆

File Edit View Insert Runtime Tools Help

COMMENT SHARE A

CODE TEXT CELL CELL CONNECT EDITING

Table of contents Code snippets Files

Copyright 2018 The TensorFlow Authors.

Predict house prices: regression

- The Boston Housing Prices dataset
  - Examples and features
  - Labels
- Normalize features
- Create the model
- Train the model
- Predict
- Conclusion

SECTION

Copyright 2018 The TensorFlow Authors.

↳ *2 cells hidden*

Predict house prices: regression

View on TensorFlow.org Run in Google Colab View source on GitHub

In a *regression* problem, we aim to predict the output of a continuous value, like a price or a probability. Contrast this with a *classification* problem, where we aim to predict a discrete label (for example, where a picture contains an apple or an orange).

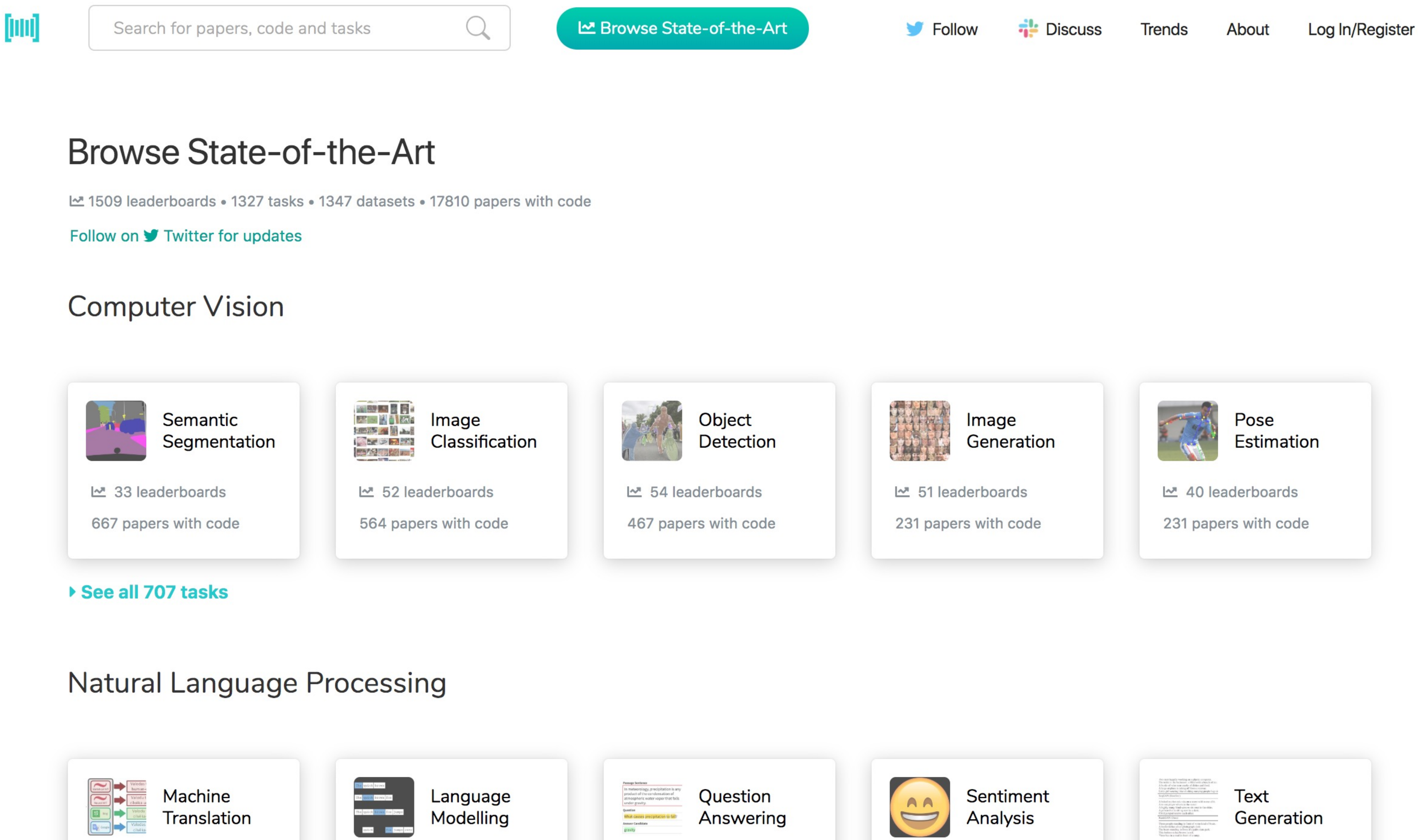
This notebook builds a model to predict the median price of homes in a Boston suburb during the mid-1970s. To do this, we'll provide the model with some data points about the suburb, such as the crime rate and the local property tax rate.

This example uses the `tf.keras` API, see this guide for details.

```
# memory footprint support libraries/code
!ln -sf /opt/bin/nvidia-smi /usr/bin/nvidia-smi
!pip install gputil
!pip install psutil
!pip install humanize
import psutil
import humanize
import os
import GPUtil as GPU
GPUs = GPU.getGPUs()
gpu = GPUs[0]
def printm():
    process = psutil.Process(os.getpid())
    print("Gen RAM Free: " + humanize.naturalsize( psutil.virtual_memory().available ), " | Proc size: "
    print("GPU RAM Free: {0:.0f}MB | Used: {1:.0f}MB | Util {2:3.0f}% | Total {3:.0f}MB".format(gpu.memo
```

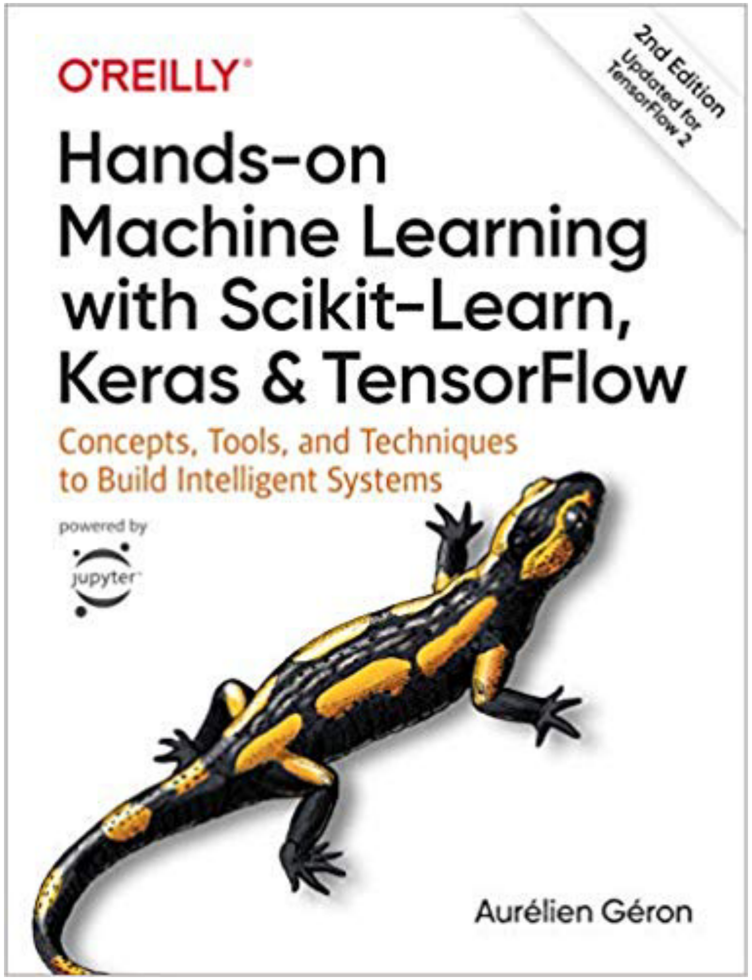
Source: https://colab.research.google.com/github/tensorflow/docs/blob/master/site/en/tutorials/keras/basic_regression.ipynb

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



https://paperswithcode.com/sota

Aurélien Géron (2019),
Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition
O'Reilly Media, 2019



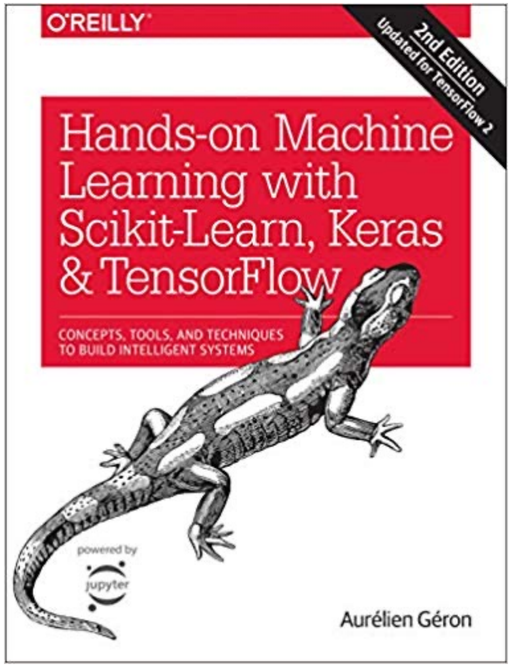
https://github.com/ageron/handson-ml2

Source: https://www.amazon.com/Hands-Machine-Learning-Scikit-Learn-TensorFlow/dp/1492032646/

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

Notebooks

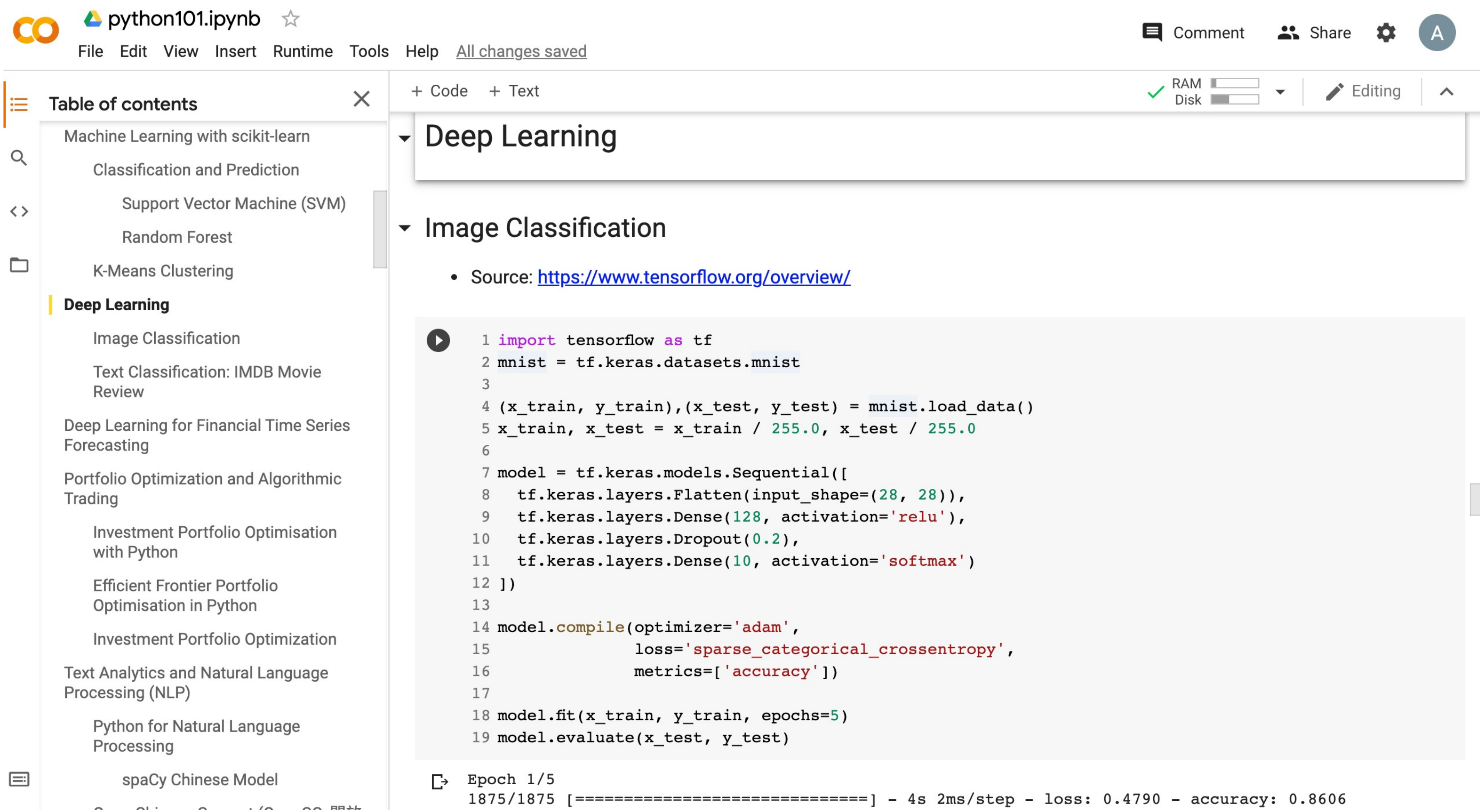
1. The Machine Learning landscape
2. End-to-end Machine Learning project
3. Classification
4. Training Models
5. Support Vector Machines
6. Decision Trees
7. Ensemble Learning and Random Forests
8. Dimensionality Reduction
9. Unsupervised Learning Techniques
10. Artificial Neural Nets with Keras
11.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



https://github.com/ageron/handson-ml2

Python in Google Colab (Python101)

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



https://tinyurl.com/aintpupython101

Summary

- Deep Learning
- Neural Networks (NN)
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)

References

- Stuart Russell and Peter Norvig (2020), Artificial Intelligence: A Modern Approach, 4th Edition, Pearson.
- Aurélien Géron (2019), Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 2nd Edition, O'Reilly Media.
- Ahmet Murat Ozbayoglu, Mehmet Ugur Gudelek, and Omer Berat Sezer (2020). "Deep learning for financial applications: A survey." Applied Soft Computing (2020): 106384.
- Omer Berat Sezer, Mehmet Ugur Gudelek, and Ahmet Murat Ozbayoglu (2020), "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019." Applied Soft Computing 90 (2020): 106181.
- Deep Learning Basics: Neural Networks Demystified, https://www.youtube.com/playlist?list=PLiaHhY2iBX9hdHaRr6b7XevZtgZRa1PoU
- Deep Learning SIMPLIFIED, https://www.youtube.com/playlist?list=PLjJh1vlSEYgvGod9wWiydumYl8hOXixNu
- 3Blue1Brown (2017), But what *is* a Neural Network? | Chapter 1, deep learning, https://www.youtube.com/watch?v=aircAruvnKk
- 3Blue1Brown (2017), Gradient descent, how neural networks learn | Chapter 2, deep learning, https://www.youtube.com/watch?v=IHZwWFHWa-w
- 3Blue1Brown (2017), What is backpropagation really doing? | Chapter 3, deep learning, https://www.youtube.com/watch?v=Ilg3gGewQ5U
- Scikit-learn – Machine Learning in Python: https://scikit-learn.org/
- TensorFlow: https://www.tensorflow.org/
- PyTorch: https://pytorch.org/
- Min-Yuh Day (2021), Python 101, https://tinyurl.com/aintpupython101