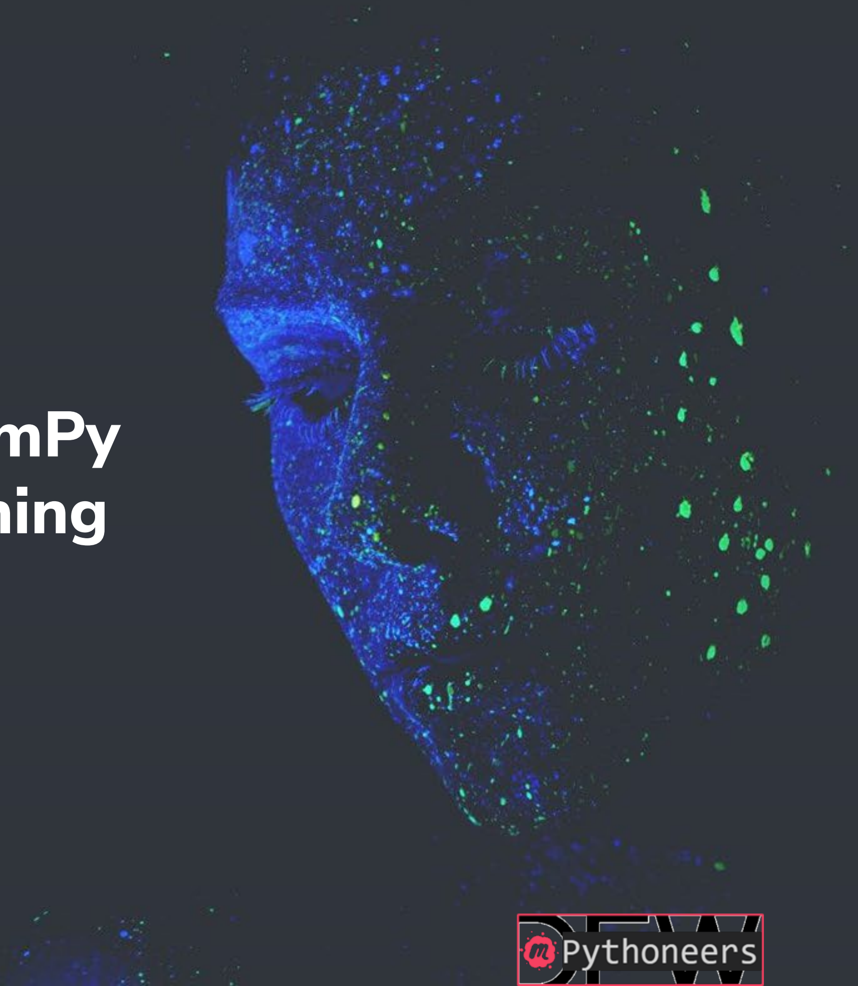
DIVERGENCE ACADEMY

Advanced Analytics: From SciPy, StatsModel & SimPy to Deep Reinforcement Learning



Drew Minkin
Lead Data Science Instructor
Nov 3 2022

Pythoneers

Topics Overview

1. The Triumverate of Advanced Analytics
2. StatsModel and Prediction
3. SciPy and Optimization
4. SimPy and Simulation
5. Tying it all Together with Reinforcement Learning

Who's Drew?

Previous Incarnations

SQL Support/Consulting 2000-06

Data Scientist 2010 – Now

5 Startups 2007-19

Analytics Architect/Dev

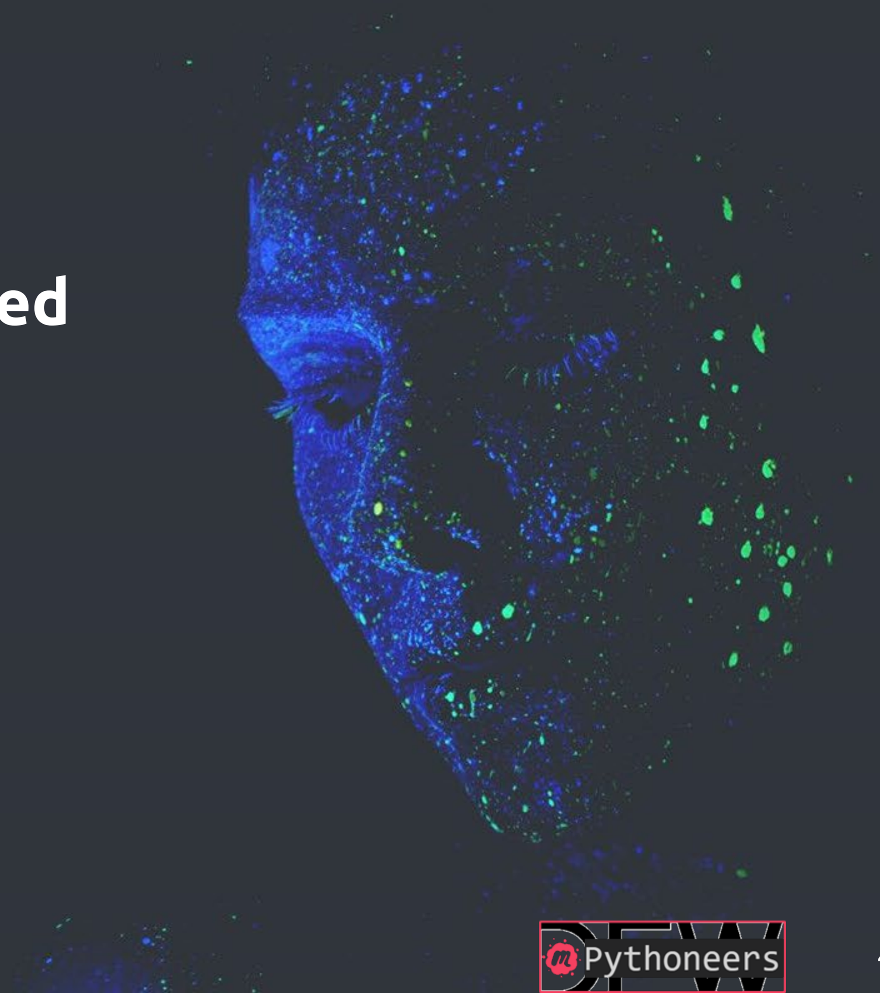
Entrepreneur

Current Incarnation

MCT, Data Science Instructor

DIVERGENCE ACADEMY

The Triumverate of Advanced Analytics



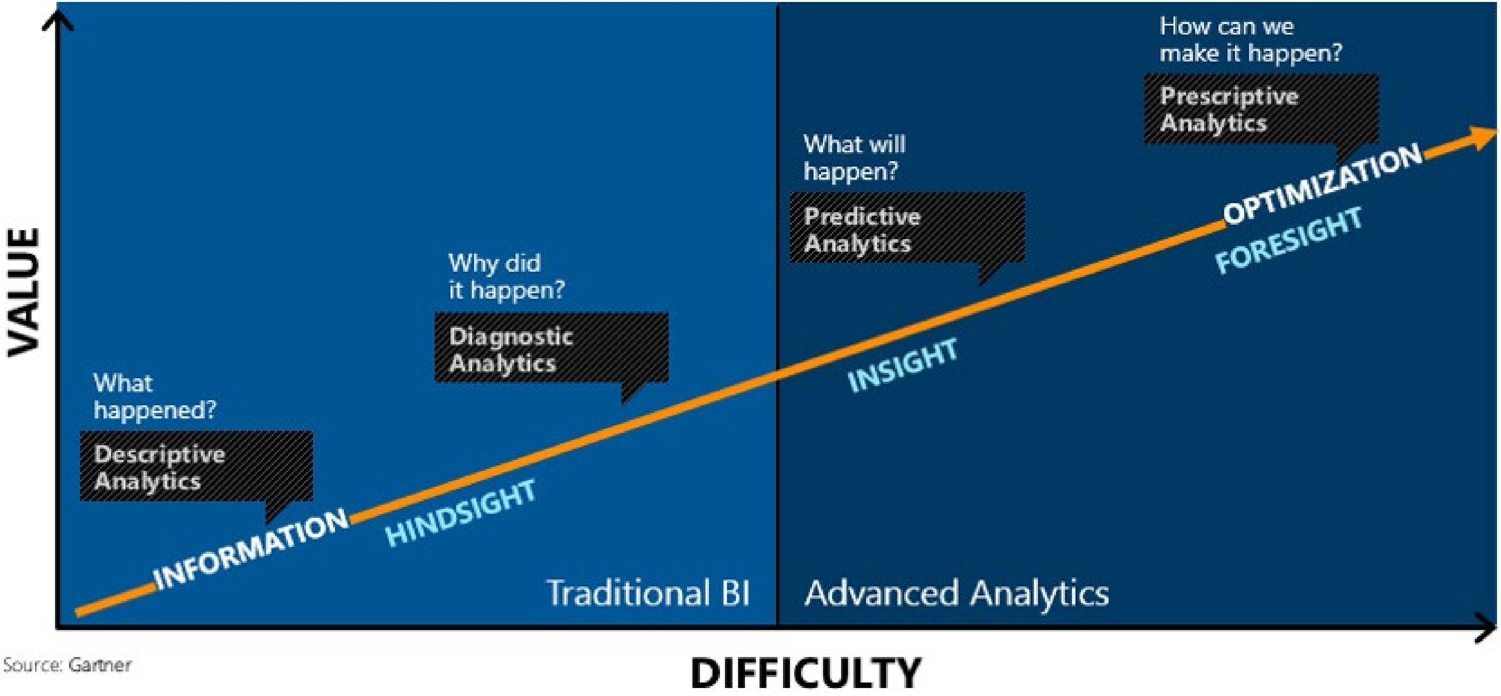
Pythoneers

Goals for Today's Session

- Know how Analytics **builds** on Business Intelligence
- Know **why** you'd build an analytic model: business payoffs
- Know **what** kinds of results you can get from analytic models
- Know **how** you'd build your own analytic model, and how to get data into your model

Adapted from https://www.solver.com/files/BAMarathon_DanielFylstra_Feb25.pptx

How Analytics Builds on Business Intelligence

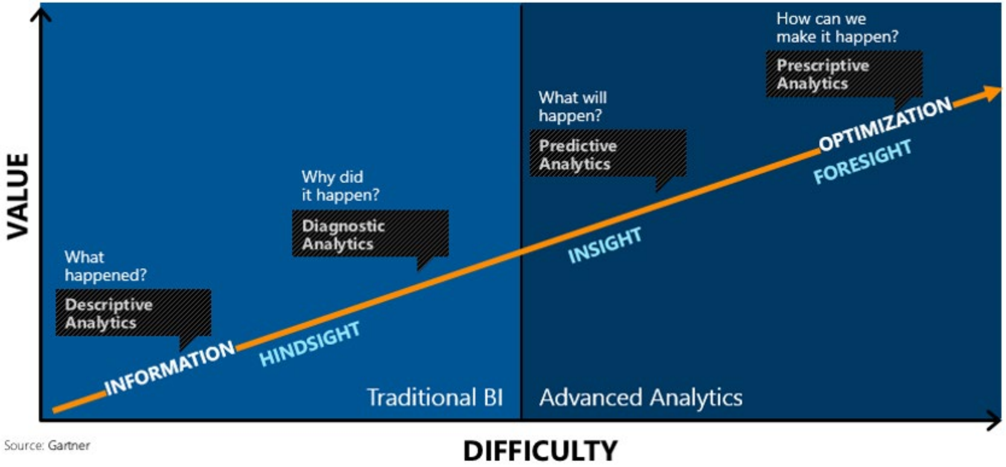


Source: Gartner

"**Analytics** are a subset of … *business intelligence*: a set of technologies and processes that use data to understand business performance … The questions that analytics can answer represent the higher-value and more proactive end of this spectrum." – *Tom Davenport, Competing on Analytics*

Adapted from https://www.solver.com/files/BAMarathon_DanielFylstra_Feb25.pptx

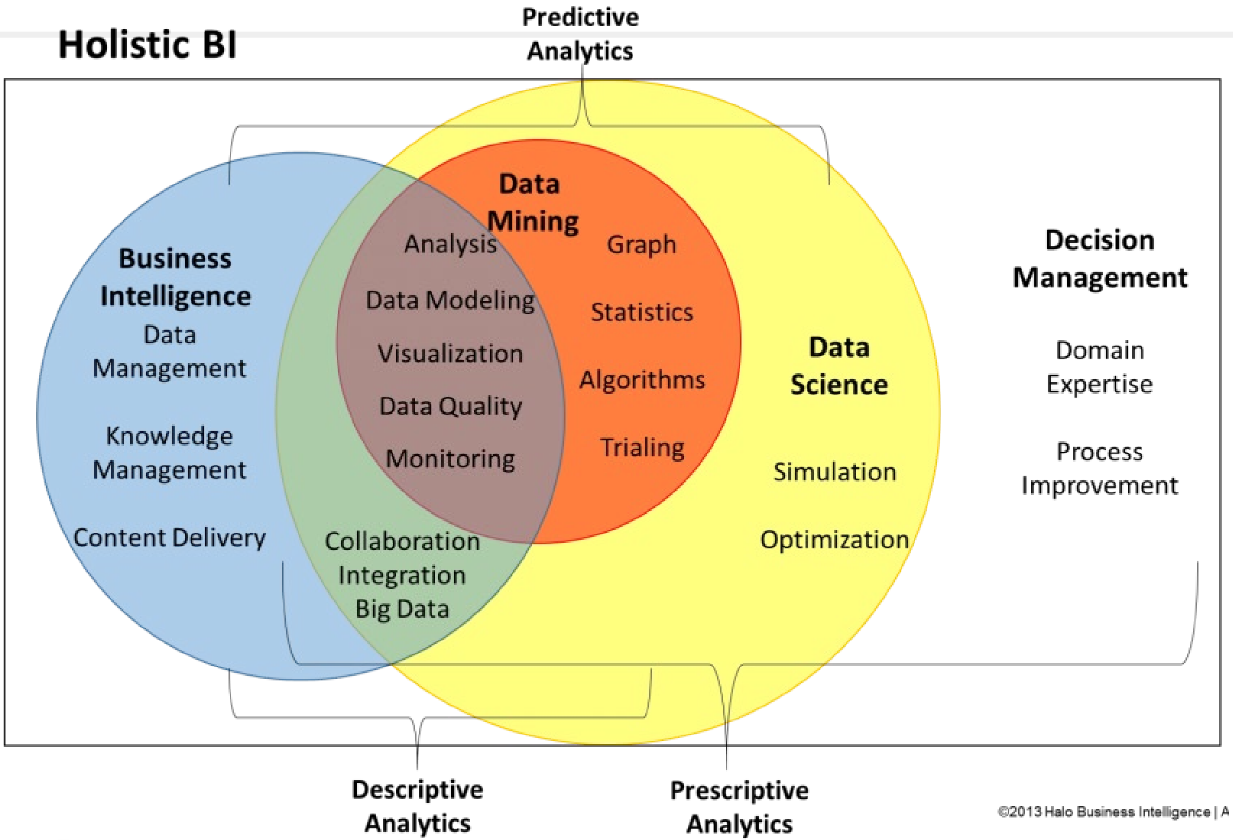
Analytics: The Three Levels



- Descriptive Analytics: Classic BI
  - Quantitative Assessment of **Past Business Results**
  - Statistics, Exploratory Data Analysis, Visualization
- Predictive Analytics
  - Quantitative Methods to **Predict New Outcomes**
  - Forecasting, Prediction, Classification, Association
- Prescriptive Analytics
  - Quantitative Methods to **Make Better Decisions**
  - Decision Trees, Monte Carlo Simulation, Optimization

Adapted from https://www.solver.com/files/BAMarathon_DanielFylstra_Feb25.pptx

Advanced Analytics Vocabulary



Analytics: The Three Analytic Models

- Results from a machine learning model:
  - Tool to *classify or predict* outcomes for new cases
  - *Assessment* of accuracy / predictive power
- Results from a simulation model:
  - *Full range* of outcomes and their likelihood
  - *Sensitivity analysis* of input parameters vs. outcomes
- Results from an optimization model:
  - *Best attainable* objective, values for decision variables
  - *Sensitivity analysis* of decision variables & constraints

Adapted from https://www.solver.com/files/BAMarathon_DanielFylstra_Feb25.pptx

Analytics: Today's Three Python Primary Packages

statsmodels

SimPy

SciPy

Why Build Analytic Models: Example Payoffs

- Two Frontline Systems Customer Examples
  - Excel model to optimally deploy 83 employees with different skill sets across 24 stations **saved** $1.9 million per year in overtime.
  - Excel simulation model showed major chemical company **why** a plant was missing goals, and **how** to solve the problem without any new investment.
- U.S. Air Force Air Logistics Center
  - C-5 Galaxy transport maintenance hub reduced turnaround time from 360 to 160 days, **saving** taxpayers $50 million and saving soldiers' lives.
- Memorial Sloan-Kettering Cancer Center
  - Optimizing radiation beams reduced side-effects of treating cancer – improving quality of life and **saving** $459 million per year on prostate cancer alone.

Adapted from https://www.solver.com/files/BAMarathon_DanielFylstra_Feb25.pptx

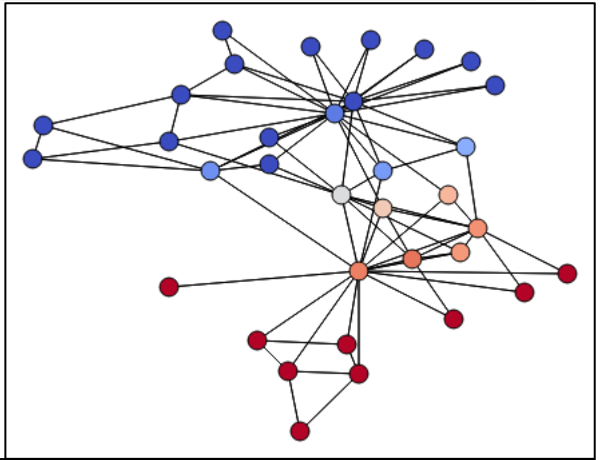
Can This Help in Your Work or Career?

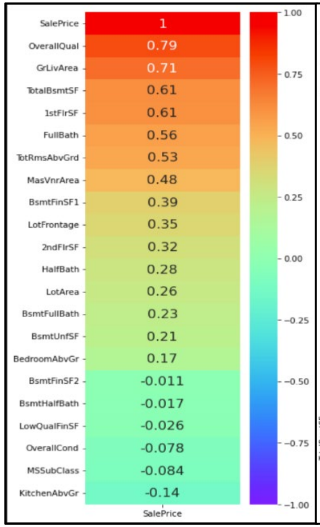
- Optimization models can deliver huge cost savings
- Simulation/risk analysis models can help avoid disaster
- But very few business analysts have the skills to do this
- If you can do this, your value to your company will rise
- Some analytic models address operations, others address strategic decisions
  - Ex. whether to build a new plant, and where to locate it
- Be prepared to present your work to senior management

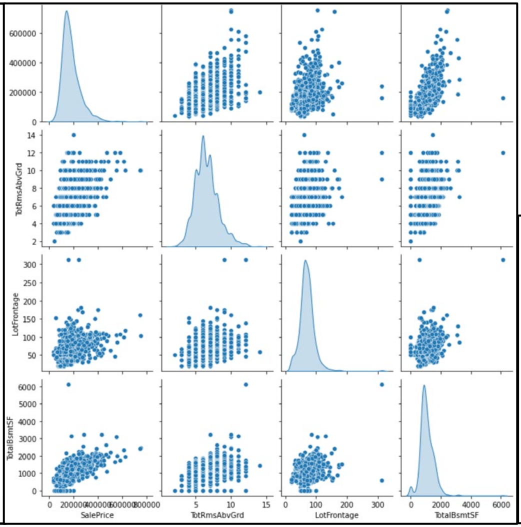
Adapted from https://www.solver.com/files/BAMarathon_DanielFylstra_Feb25.pptx

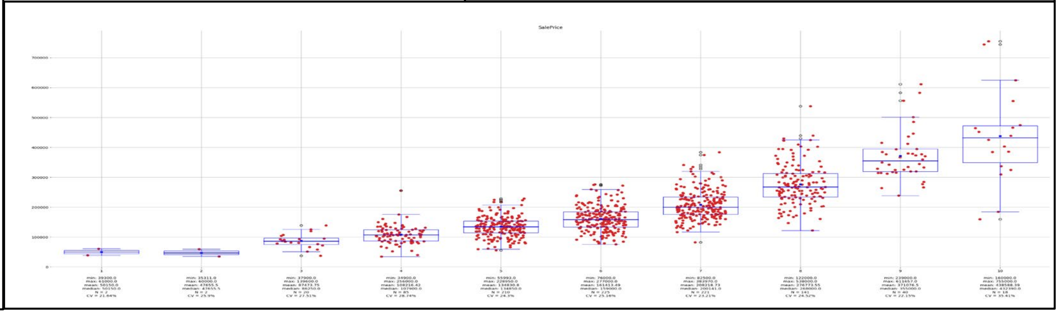
Descriptive Analytics: Before You Introduce Error

- Data access / shaping – pandas
- Visualization – matplotlib,seaborn, sknetwork
- Linear Algebra/Statistical Functions - numpy, scipy









Predictive Analytics: Beating Random

- Key tasks: Data shaping, applying predictive models
- Data mining algorithms "fit" analytic model to past data
- Trained/fitted models are applied to newly arriving data
    - **Classify**: ex. Good/Poor credit risk, Likely/Unlikely to churn
    - **Predict**: ex. stock price, house price, exchange rate
    - **Forecast** a *time series*: ex. next sales from past sales history
    - **Associate**: ex. People who bought this item also bought…
- Tools: sklearn, statsmodels, pycaret, tpot, tensorflow, pytorch, keras, onnx

Adapted from https://www.solver.com/files/BAMarathon_DanielFylstra_Feb25.pptx

Predictive Anlaytics: What You Need, How You Do It

- What You Need: Tools to
  - **Access** / shape data, explore / visualize data
  - **Train** / “fit” models to data: machine learning
  - **Validate** model results: statistics, Lift / ROC curves

- How You Do It
  - Data “wrangling” / cleaning is usually the first step
  - Use feature selection to identify variables that matter
  - Try multiple algorithms: Regression, trees, neural nets

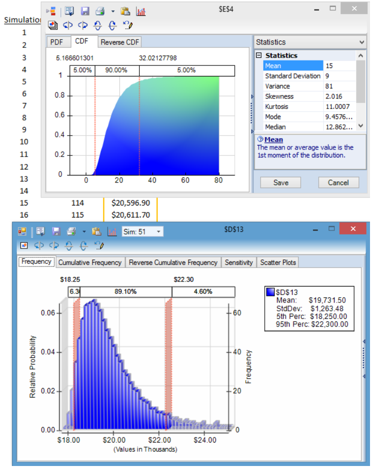
Advanced Analytics: Optimization, Simulation

- Key task: **Create a model** – A person (you) must do this
  - Model must capture essential features of the business situation
  - Larger models often get their data from BI / Descriptive Analytics
  - A "What If" model is the starting point – Excel is a natural tool!
- Given an appropriate model, we can:
  - Ask "What are ***all* the possible** outcomes?" – simulation/risk analysis
  - Ask "What's the ***best* outcome** we can achieve?" – optimization

Adapted from https://www.solver.com/files/BAMarathon_DanielFylstra_Feb25.pptx

Simulation: What You Need, How You Do It

- What You Need: Tools to
    - Create a **“what if” model**, calculating results of interest
    - Define **probability distributions** for uncertain inputs
    - Run Monte Carlo simulation, create statistics and charts

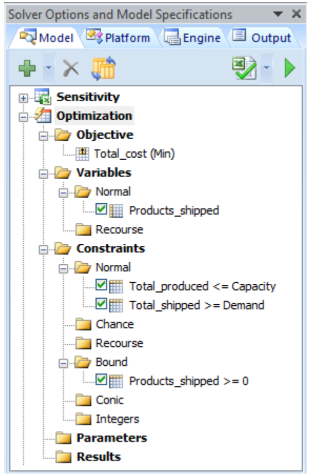


- How You Do It
    - Define distributions by fitting data, or industry practice
    - Define dependence among inputs: corr. matrices, copulas
    - Run simulation, or multiple simulations with parameters
    - **Assess** and think about results: stats, histograms, scatterplots

Optimization: What You Need, How You Do It

- What You Need: Tools to
  - Create a **“what if” model**, calculating results of interest
  - Define **decision variables** for inputs under your control
  - Define **constraints** and an **objective** to max / minimize
  - Run an optimization for optimal values, sensitivity analysis

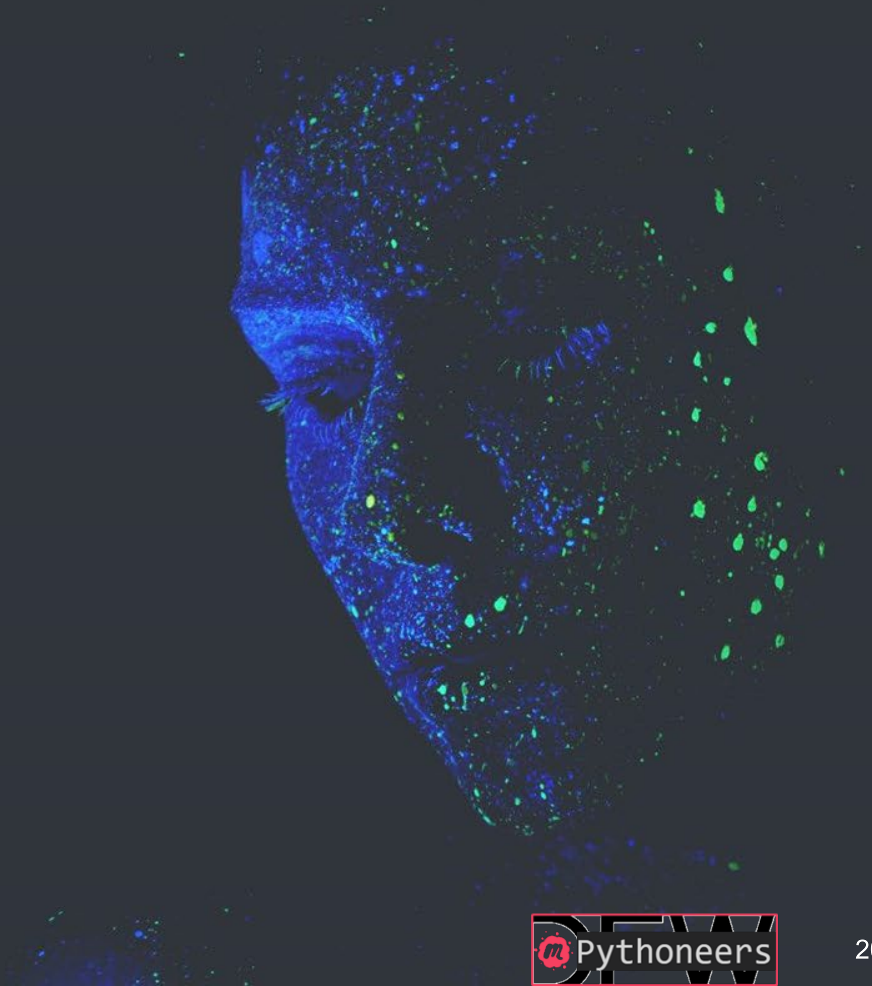
- How You Do It
  - Define constraints for limited resources, physical conditions, policies
  - Understand dependence between outputs and inputs: linear / nonlinear
  - Run optimization, or multiple optimizations with parameters you vary
  - **Assess** and think about results: understand “dual values,” sensitivity



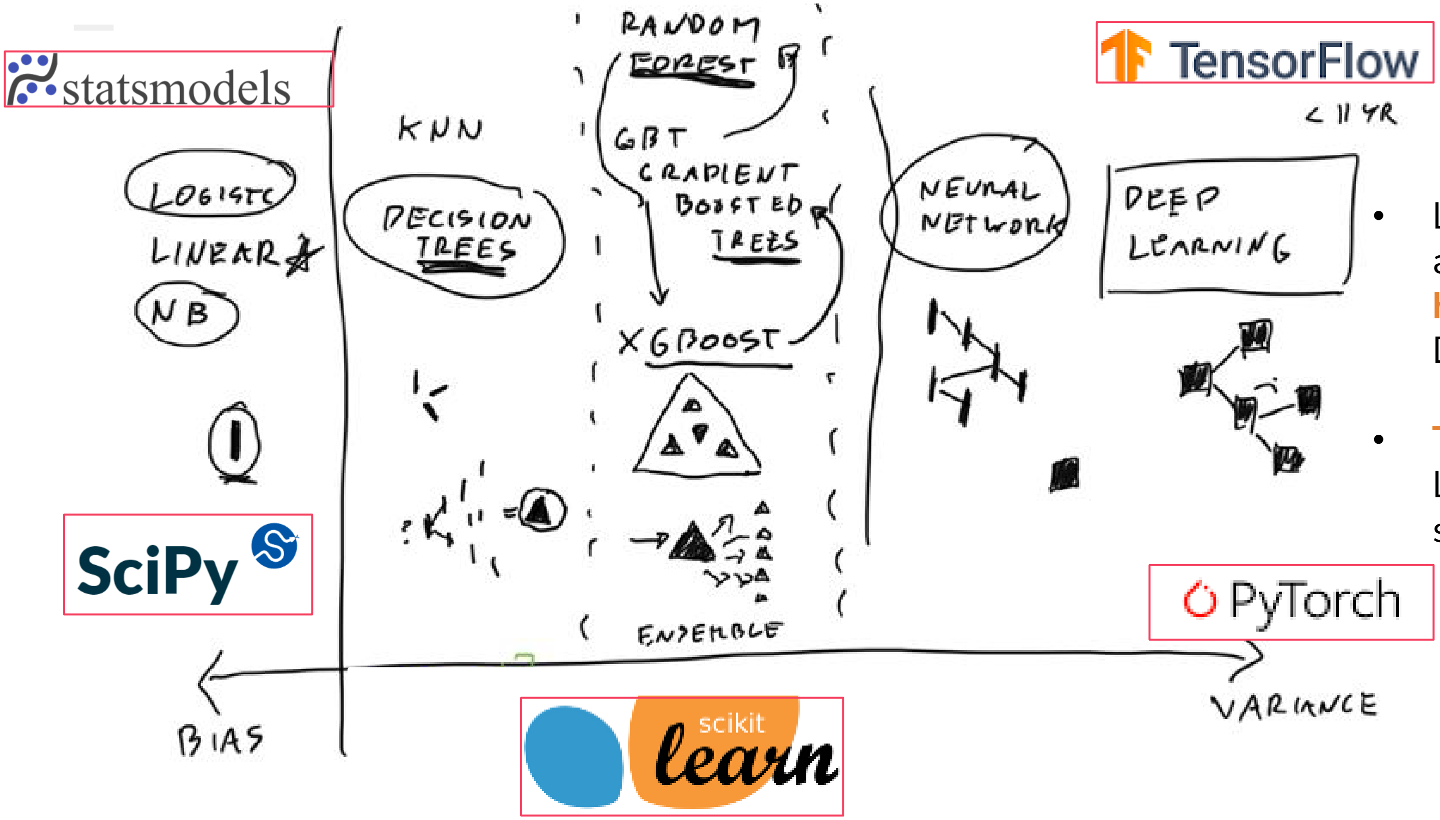
Can This Help in Your Work or Career?

- Optimization models can deliver huge cost savings
- Simulation/risk analysis models can help avoid disaster
- But very few business analysts have the skills to do this
- If you can do this, your value to your company will rise
- Some analytic models address operations, others address strategic decisions
    - Ex. whether to build a new plant, and where to locate it
- Be prepared to present your work to senior management

A Few More Notes on Predictive Analytics



Machine Learning Algorithm Complexity & Context



- LR, NN and other algorithms part of **hierarchical layers** of DL model
- **Transfer Learning** Learning models on same type of data

Machine Learning Process

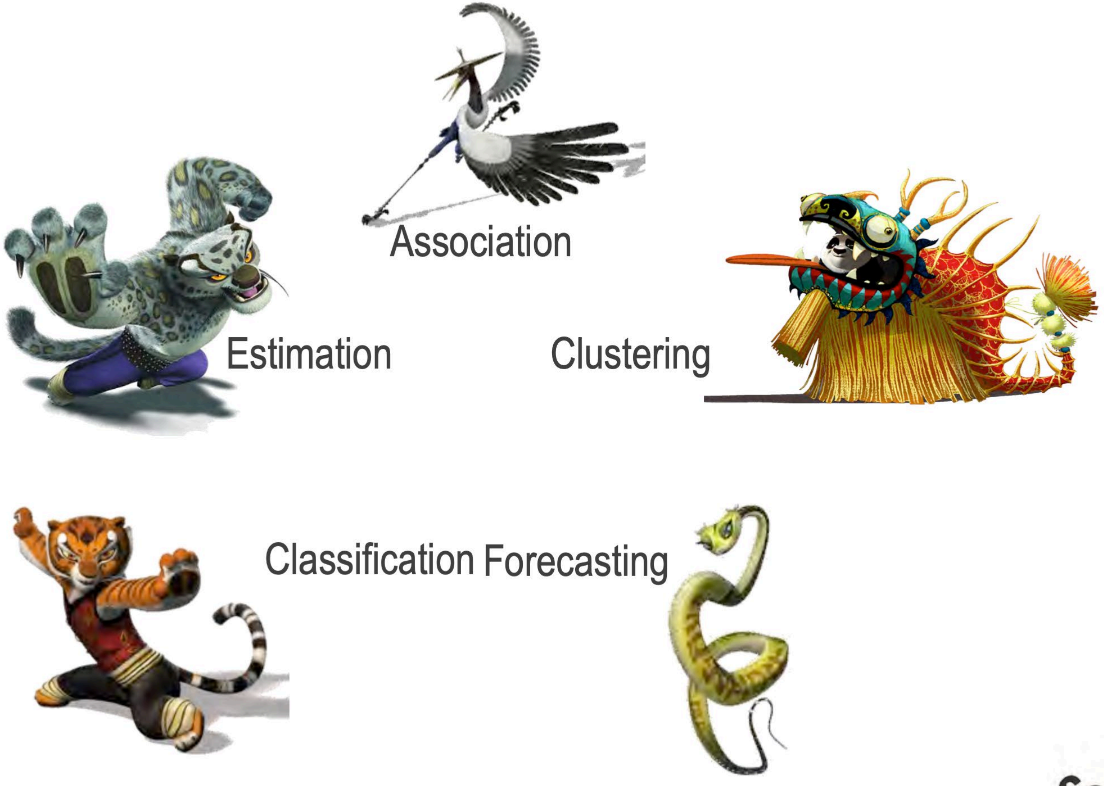
Model Production

- Data Gathering
- Feature Engineering
- Sampling and Splitting
- Modeling
- Validating
- Scoring

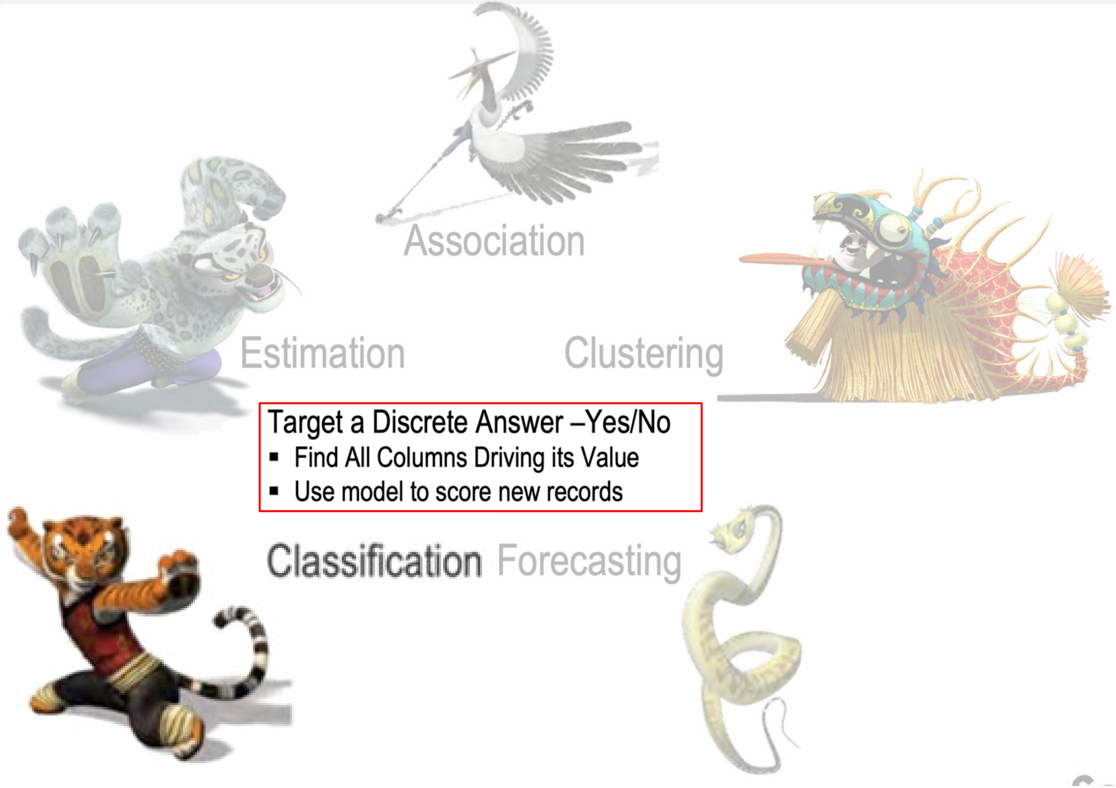
- Drift Management
- Monitor Live Scoring
- Pipeline Development
- Production Gap Analysis
- Business Alignment

Model Consumption

Machine Learning Modeling Families

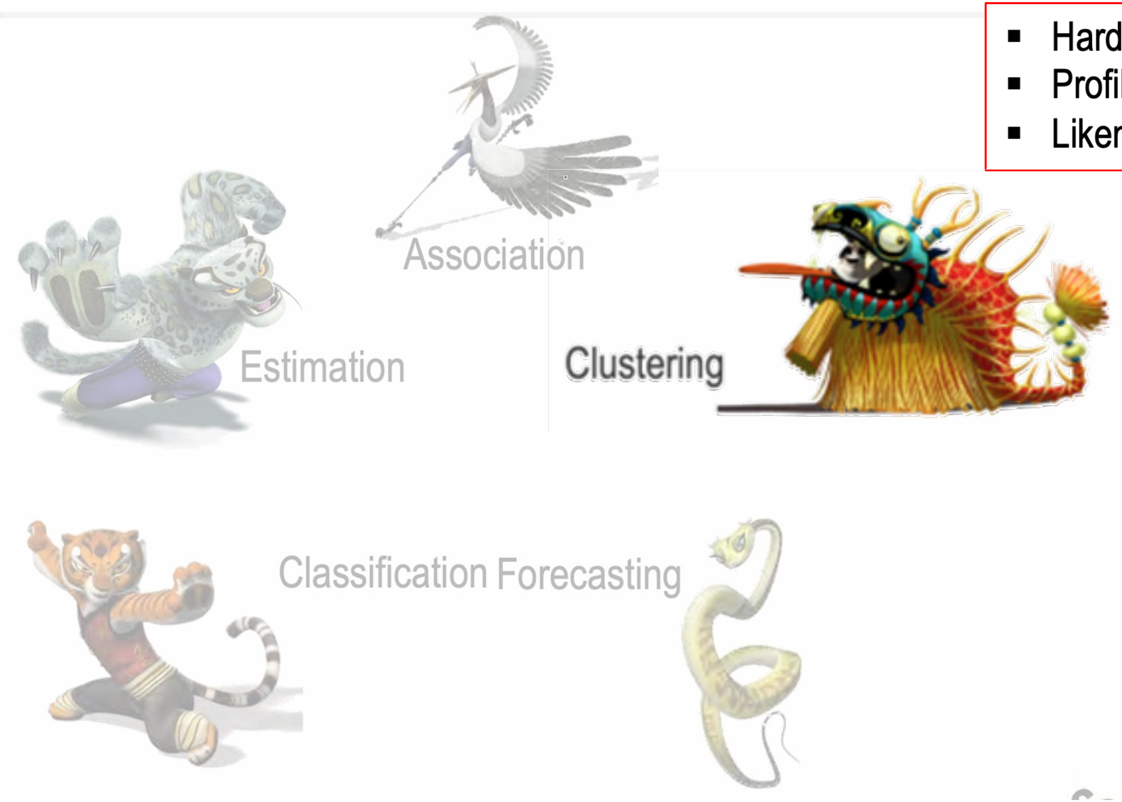


Machine Learning Modeling Families

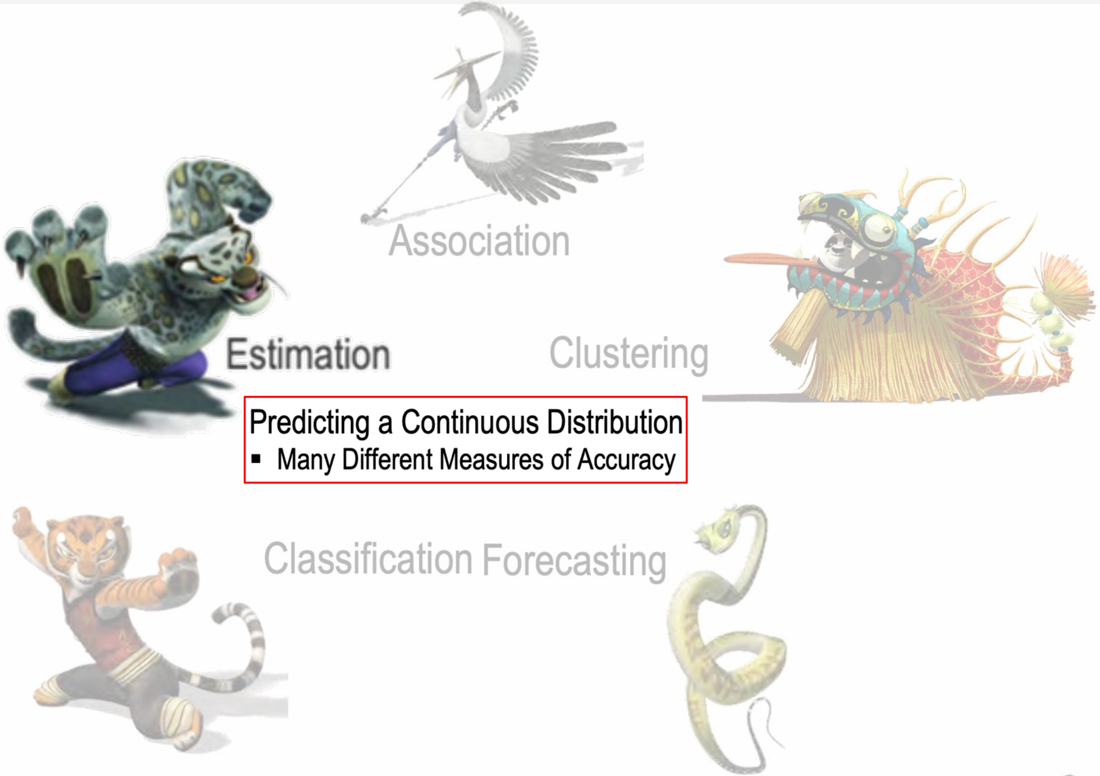


Machine Learning Modeling Families

- Hard and Soft Groupings
- Profiles of Subgroups
- Likenesses and Differences



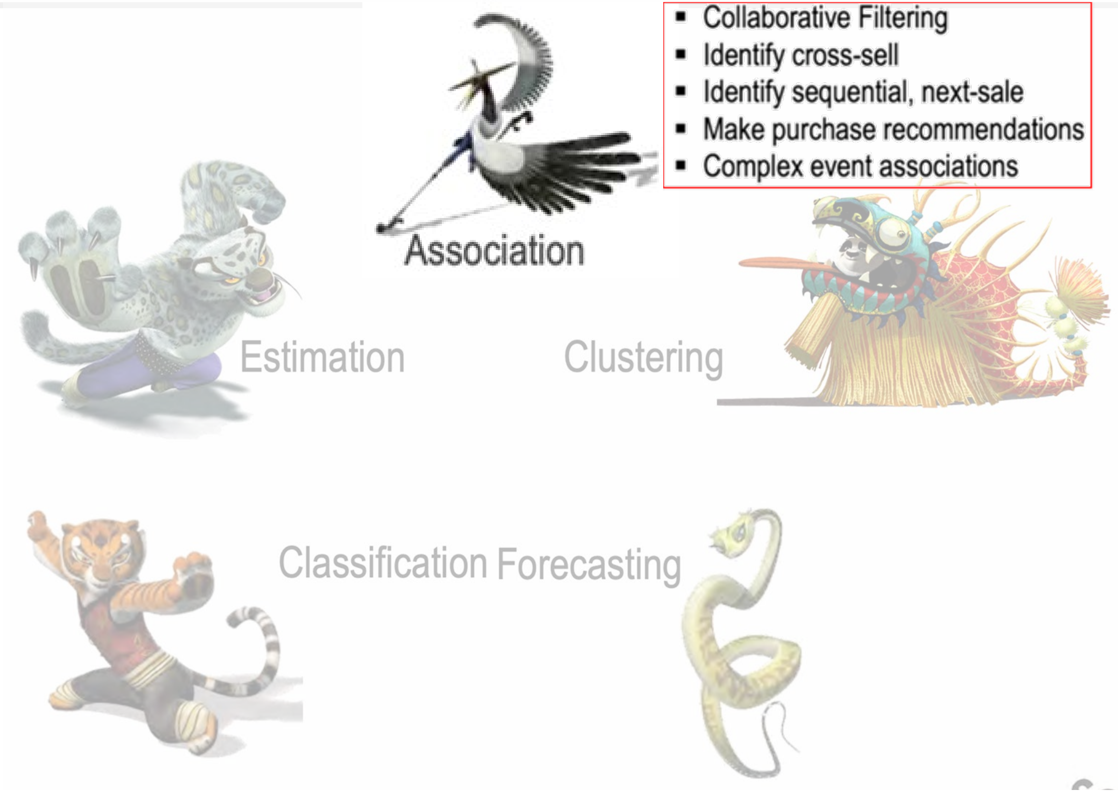
Machine Learning Modeling Families



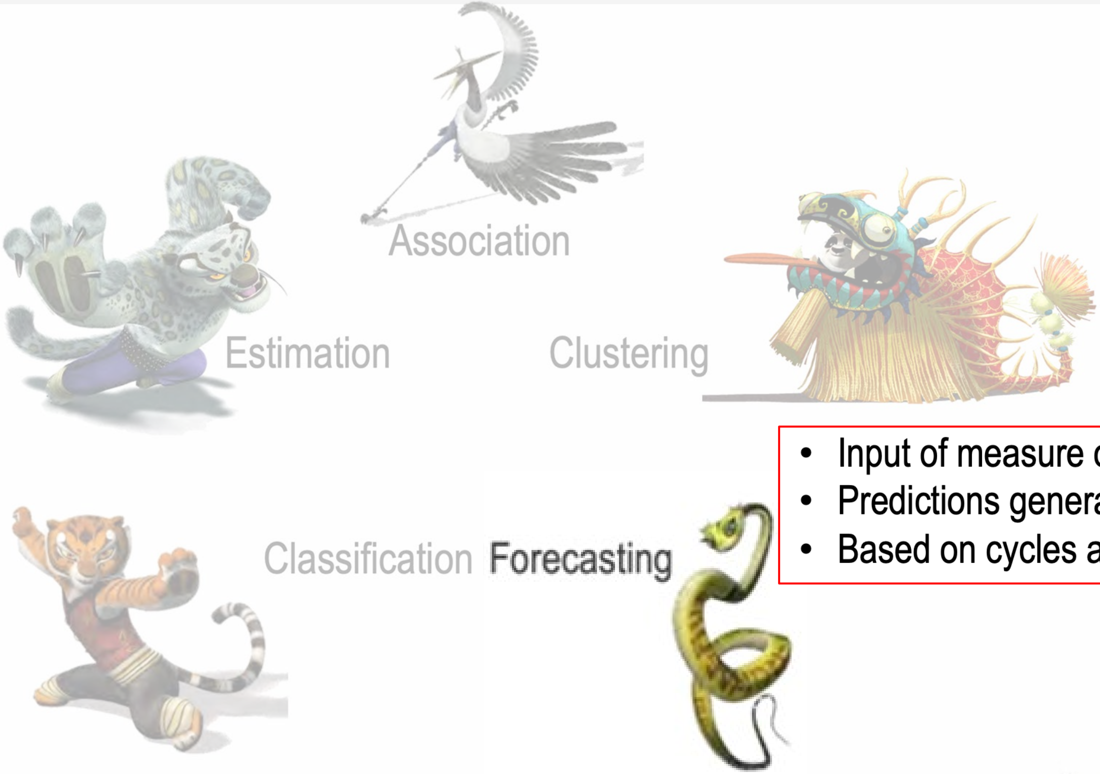
Predicting a Continuous Distribution

- Many Different Measures of Accuracy

Machine Learning Modeling Families



Machine Learning Modeling Families



- Input of measure over time and related series
- Predictions generated for short term trends
- Based on cycles and events

Statsmodels in a nutshell

- statsmodels.api
  - models and methods.
- statsmodels.tsa.api
  - Time-series models and methods. Canonically imported using import statsmodels.tsa.api as tsa.
- statsmodels.formula.api
  - A convenience interface for specifying models using formula strings and DataFrames

Time Series Algorithm Families

- Basic
  - Autoregression (AR)
  - Vector Autoregression (VAR)
- Moving Average
  - Autoregressive Moving Average (ARMA)
  - Autoregressive Integrated Moving Average (ARIMA)
  - Seasonal (SARIMA) with Exogenous Regressors (SARIMAX)
  - Vector (VARMA) with Exogenous Regressors (VARMAX)

Time Series Algorithm Families

- Smoothing
  - Simple Exponential Smoothing (SES)
  - Holt Winter's Exponential Smoothing (HWES)

Time Series Algorithm Families

- Smoothing
  - Simple Exponential Smoothing (SES)
    - Exponentially weighted linear function
  - Holt Winter's Exponential Smoothing (HWES)
    - Triple SES
    - Supports trends and seasonality

AutoRegression

- Next step in the sequence as a linear function of the observations at prior time steps.
- AR(p) Notation
  - AR(1) is a first-order autoregression model.
- Univariate time series without trend and seasonal components.

Vector AutoRegression

- Multivariate time series without trend and seasonal components.
- VAR(p) notation
  - VAR(1) is a first-order vector autoregression model.

Autoregressive Moving Average (ARMA)

- It combines both Autoregression (AR) and Moving Average (MA) models.
- Notation
  - ARMA(p, q).
  - MA(q) is order of MA model
  - can be used to develop AR or MA models.
- Univariate time series without trend and seasonal components)

Autoregressive Integrated Moving Average (ARIMA)

- Integrates both Autoregression (AR) and Moving Average (MA) models.
- Notation
  - ARIMA(p,d, q)
  - d =s the number of deltas required for best fit
- Univariate time series without trend and seasonal components)

Seasonal ARIMA with Exogenous Regressors (SARIMAX)

- Basic
  - SARIMA(p, d, q)(P, D, Q)m
  - p,d,q – ARIMA base
  - P,D,Q -
  - m = number of time steps in each period
- AR, MA, ARMA and ARIMA models.
- Exogenous regressors upport multivariate analysis

DIVERGENCE ACADEMY

DEMO
StatsModel & Power BI

Pythoneers

DIVERGENCE ACADEMY

Optimization and SciPy

Pythoneers

SciPy in a Nutshell

- SciPy is a library of algorithms and mathematical tools built to work with NumPy arrays.
    - linear algebra- scipy.linalg
    - statistics- scipy.stats
    - optimization- scipy.optimize
    - sparse matrices- scipy.sparse
    - signal processing- scipy.signal

https://web.stanford.edu/~schmit/cme193/lec/lec5.pdf

SciPy Statistics & Sparse

- scipy.sparse
  - Sparse matrix classes: CSC, CSR, etc. Functions to build sparse matrices
  - sparse.linalg module for sparse linear algebra
  - sparse.csgraph for sparse graph routines
- scipy.statistics
  - Mean, median, mode, variance, kurtosis
  - Pearson correlation coefficient
  - Hypothesis tests (ttest, Wilcoxon signed-rank test, Kolmogorov-Smirnov) Gaussian kernel density estimation

https://web.stanford.edu/~schmit/cme193/lec/lec5.pdf

SciPy IO & Signal

- scipy.sparse
  - Sparse matrix classes: CSC, CSR, etc. Functions to build sparse matrices
  - sparse.linalg module for sparse linear algebra
  - sparse.csgraph for sparse graph routines
- scipy.statistics
  - Mean, median, mode, variance, kurtosis
  - Pearson correlation coefficient
  - Hypothesis tests (ttest, Wilcoxon signed-rank test, Kolmogorov-Smirnov) Gaussian kernel density estimation

https://web.stanford.edu/~schmit/cme193/lec/lec5.pdf

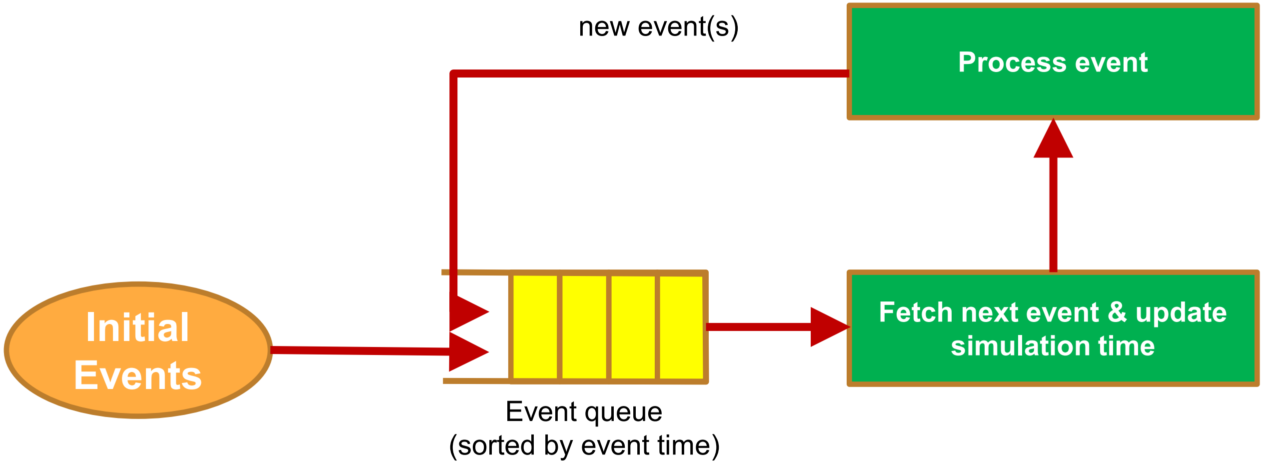
DEMO
SciPy –Clustering & Optimization

DIVERGENCE ACADEMY

Simulation And Simpy

Discrete-Event Simulation

- Simulated operations are performed as a discrete sequence of events in time
- "Time" stops during event processing



SimPy Simulator

- Process-based discrete-event simulator written in Python
- Simulated processes are defined as Python coroutines (i.e., generators)

SimPy Example

- The following code simulates cars arriving at three toll booths at random points in time
    - Assuming cars' inter-arrival times are exponentially distributed, with the average inter-arrival time of 3

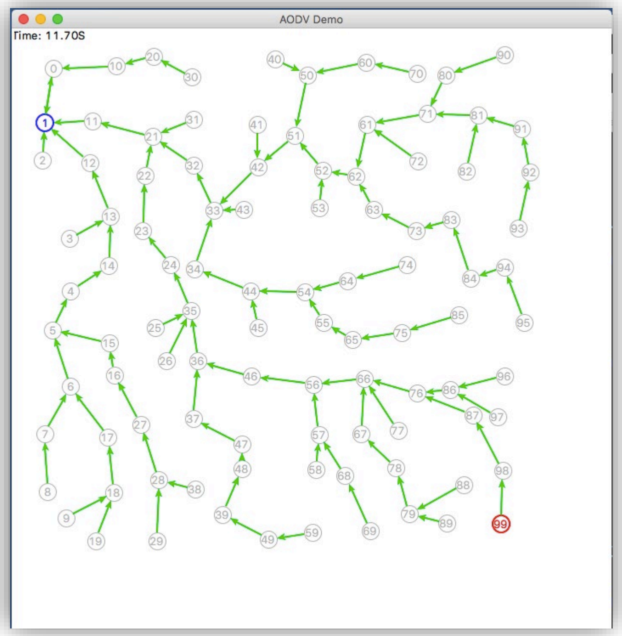
```python
import numpy as np
import simpy

def booth(name,env):
    count = 0
    while True:
        yield env.timeout(np.random.exponential(30))
        count += 1
        print(f"At {env.now:3.0f} seconds, car #{count} arrives at {name}")

env = simpy.Environment()
env.process(booth("Booth 1",env))
env.process(booth("Booth 2",env))
env.process(booth("Booth 3",env))
env.run(until=300)
```

Introduction to WsnSimPy

- WSN simulator based on SimPy
- Implements basic Node model and simple collision-free node-to-node communication



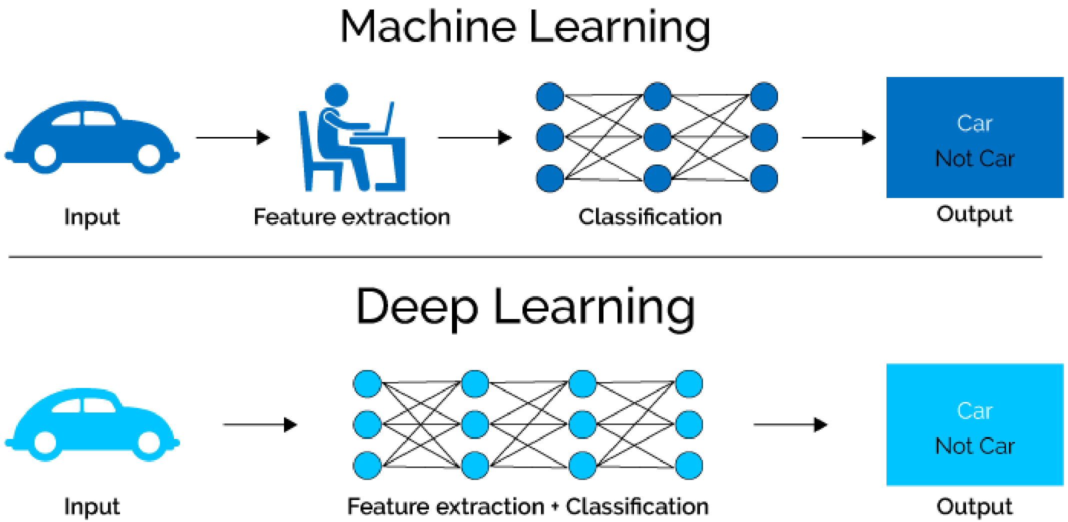
DIVERGENCE ACADEMY

DEMO
WsnSimpy

DIVERGENCE ACADEMY

Putting It All Together: Deep Learning & Reinforcement Learning

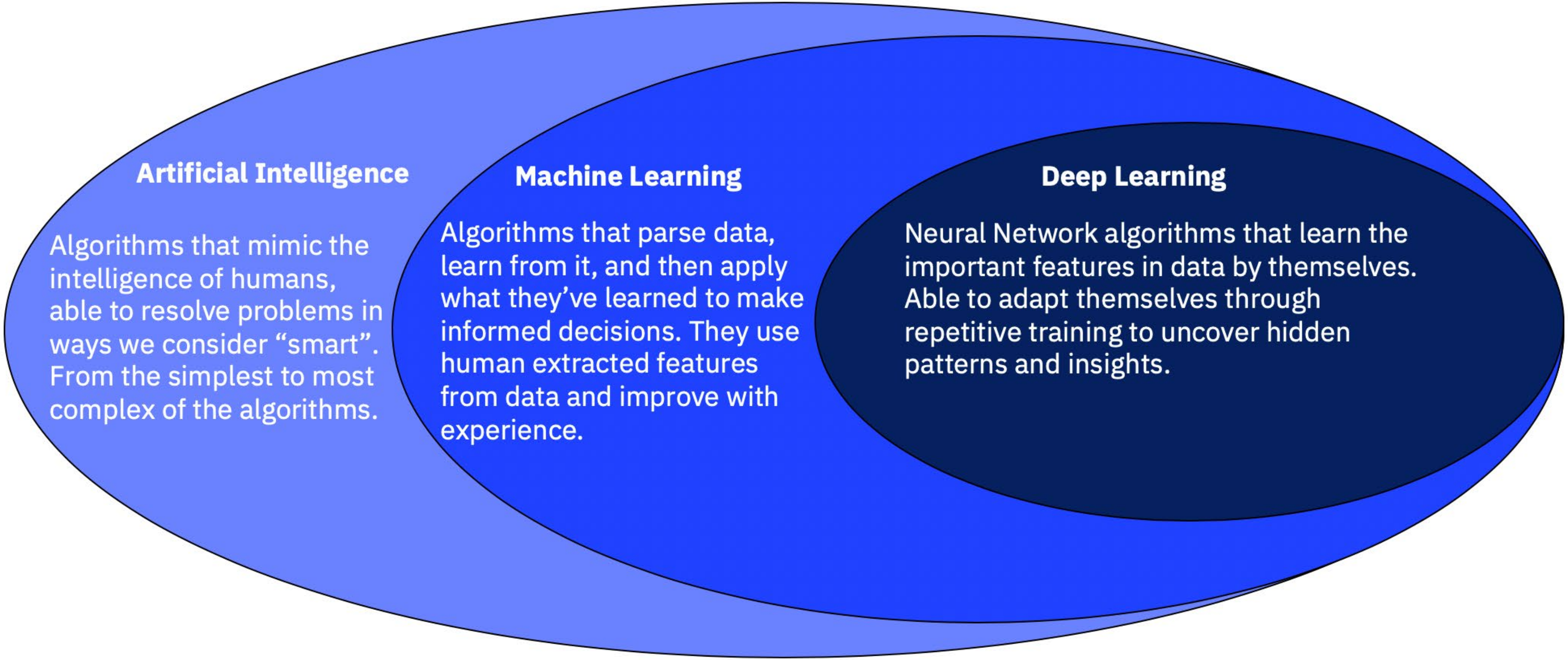
Machine Learning vs Deep Learning



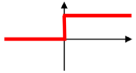
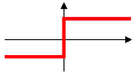
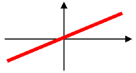
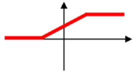
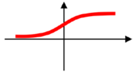
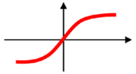
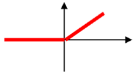
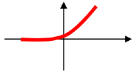
- A more complex **representation** of data
- .
- Processing of **complex feature spaces**
- **Feature Engineering baked into Models**
- **Deep** layering of algorithm, not wide.

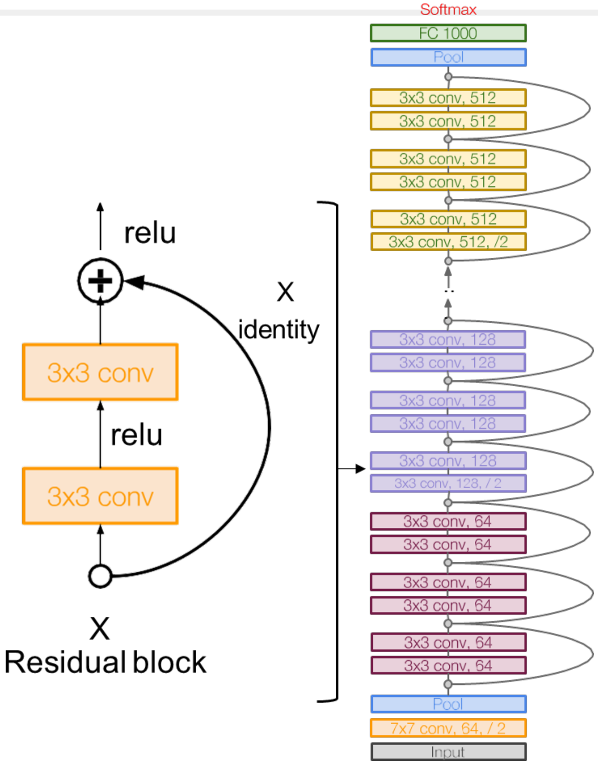
https://www.xenonstack.com/blog/static/public/uploads/media/machine-learning-vs-deep-learning.png

Deep Learning, Machine Learning, and AI



Deep Learning and Activation Functions

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^z)$	Multi-layer Neural Networks	

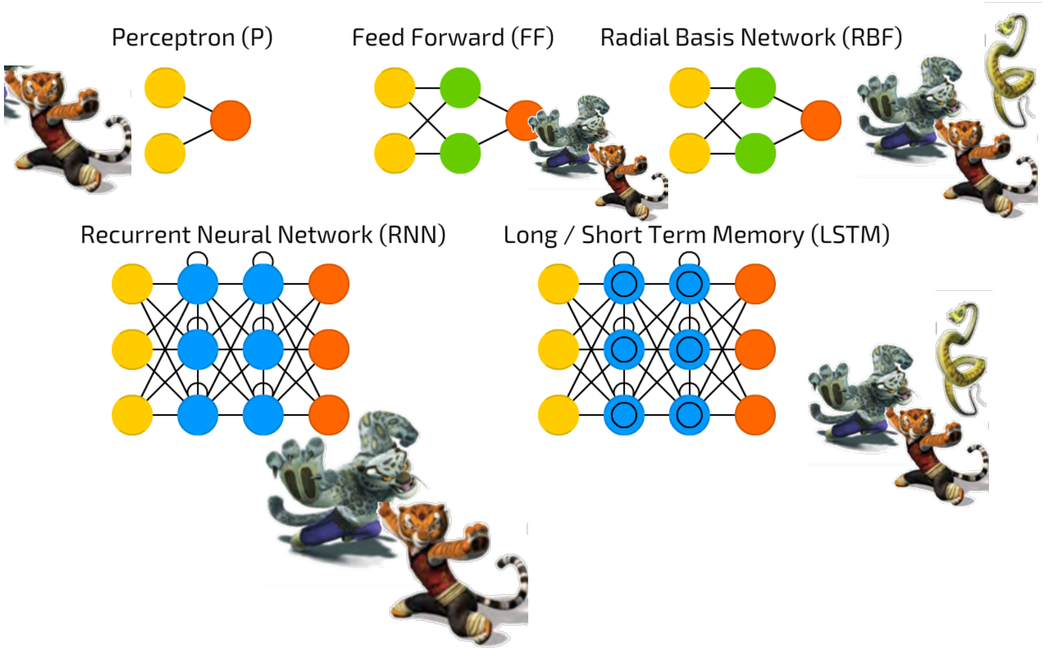


https://www.pinterest.com/pin/672232681856671318/

Deep Learning General Model Structure Examples

Basic Structures

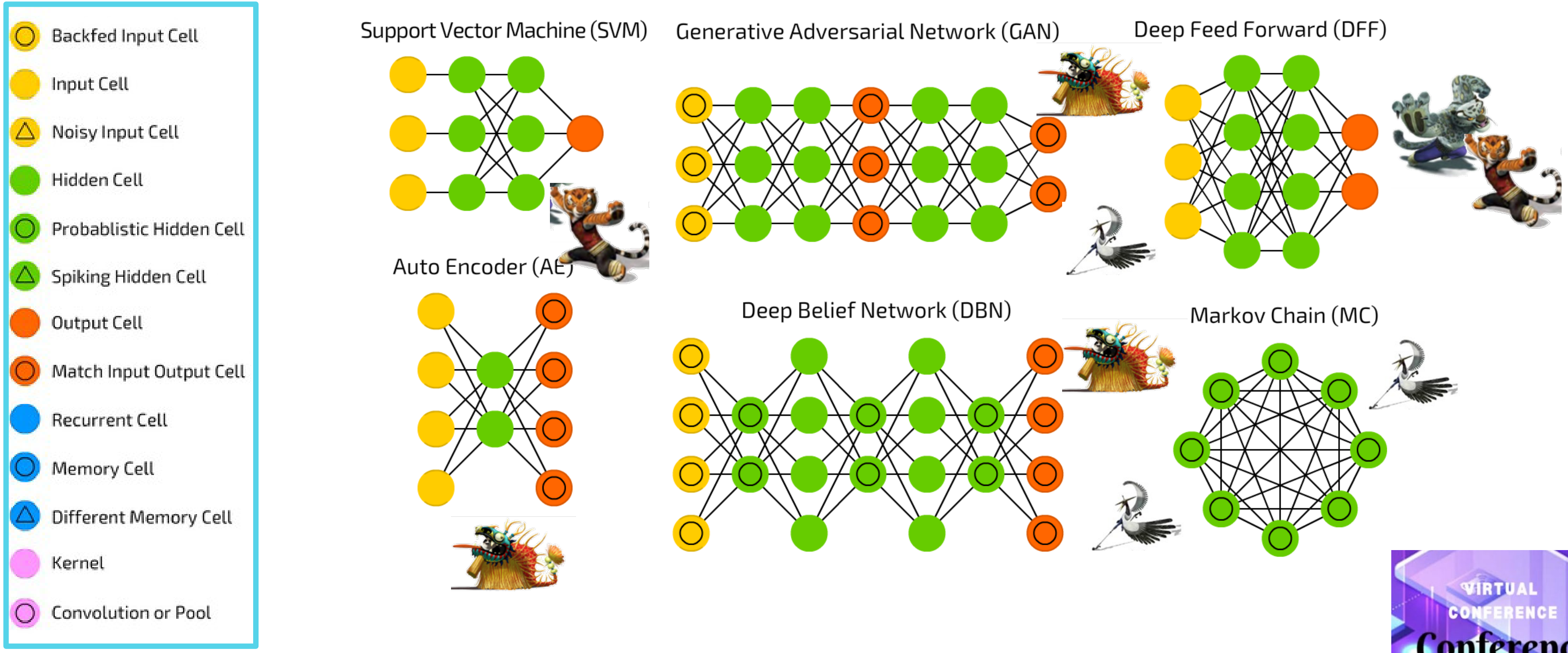
- Backfed Input Cell
- Input Cell
- Noisy Input Cell
- Hidden Cell
- Probablistic Hidden Cell
- Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- Different Memory Cell
- Kernel
- Convolution or Pool



https://www.kaggle.com/getting-started/151100

Deep Learning General Model Structure Examples

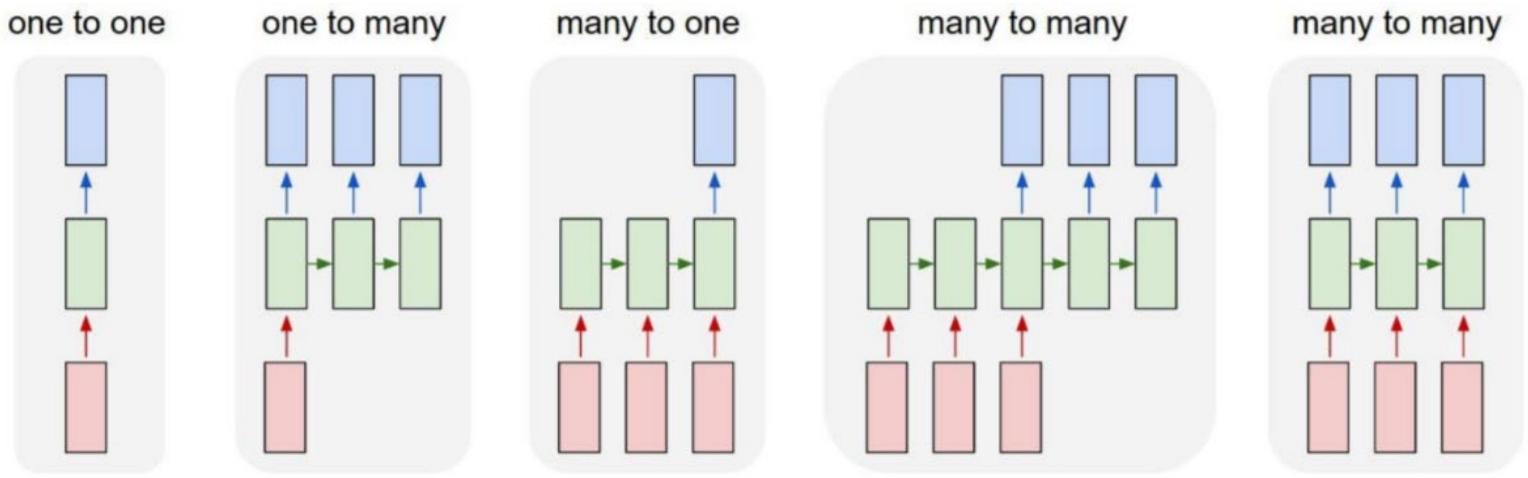
Next Generation Complexity



https://www.kaggle.com/getting-started/151100

Deep Learning General Model Structure Examples

Sequence to Sequence (Seq2Seq)



Fixed-sized input to fixed-sized output (e.g. image classification)

Sequence output (e.g. image captioning takes an image and outputs a sentence of words).

Sequence input (e.g. sentiment analysis where a given sentence is classified as expressing positive or negative sentiment).

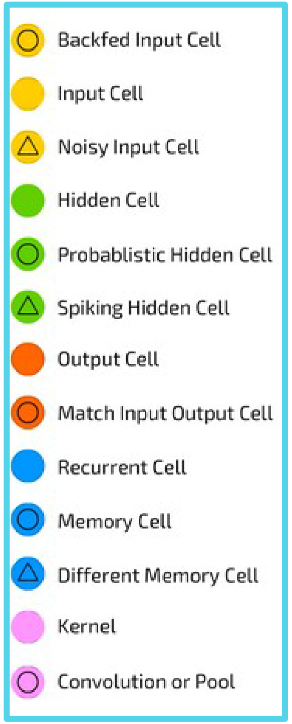
Sequence input and sequence output (e.g. Machine Translation: an RNN reads a sentence in English and then outputs a sentence in French)

Synced sequence input and output (e.g. video classification where we wish to label each frame of the video)

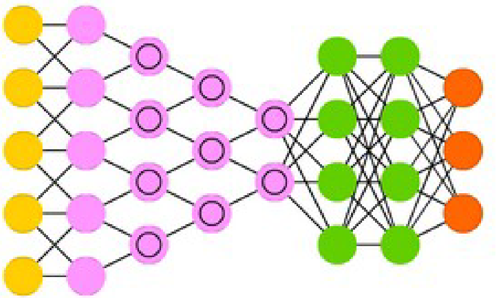
https://mett29.github.io/posts/2019/12/seq2seq_and_attention/

Deep Learning General Model Structure Examples

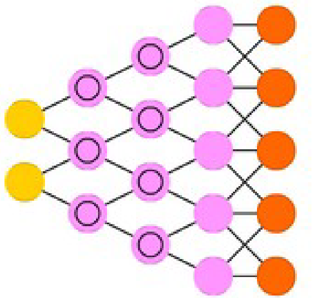
Next Generation Complexity



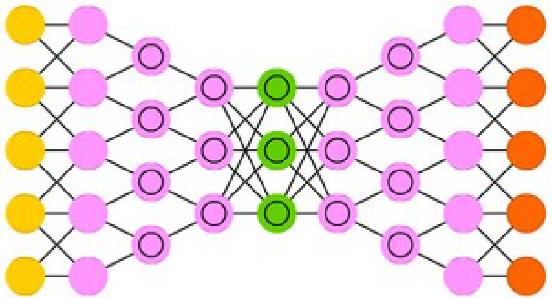
Deep Convolutional Network (DCN)



Deconvolutional Network (DN)



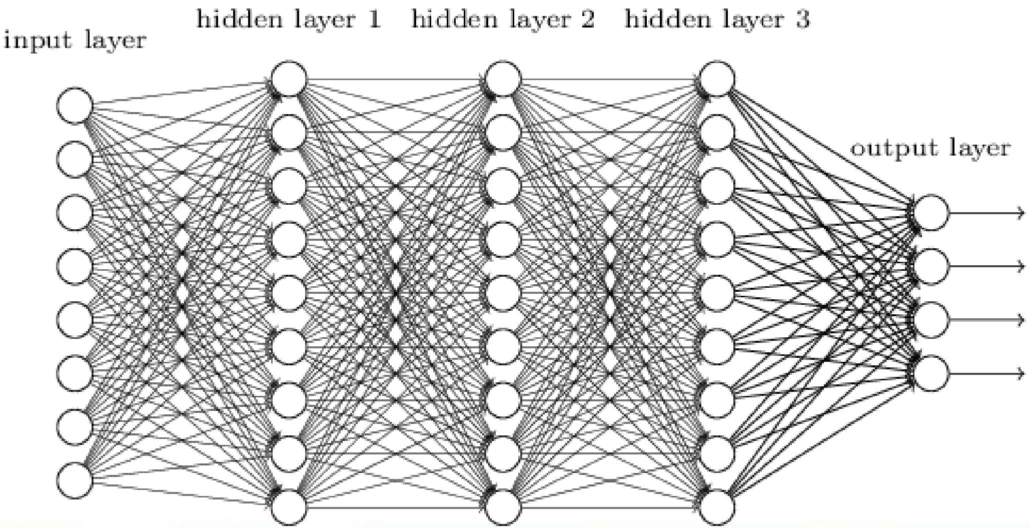
Deep Convolutional Inverse Graphics Network (DCIGN)



https://www.kaggle.com/getting-started/151100

Why Simple Neural Networks don't cut it

- We know it is good to learn a small model.
- From this fully connected model, do we really need all the edges?
- Can some of these be shared?

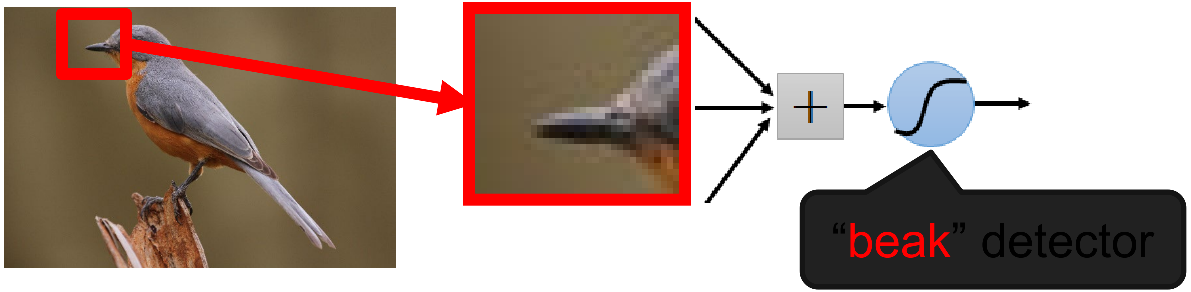


https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture5CNN.ppt

Consider learning an image:

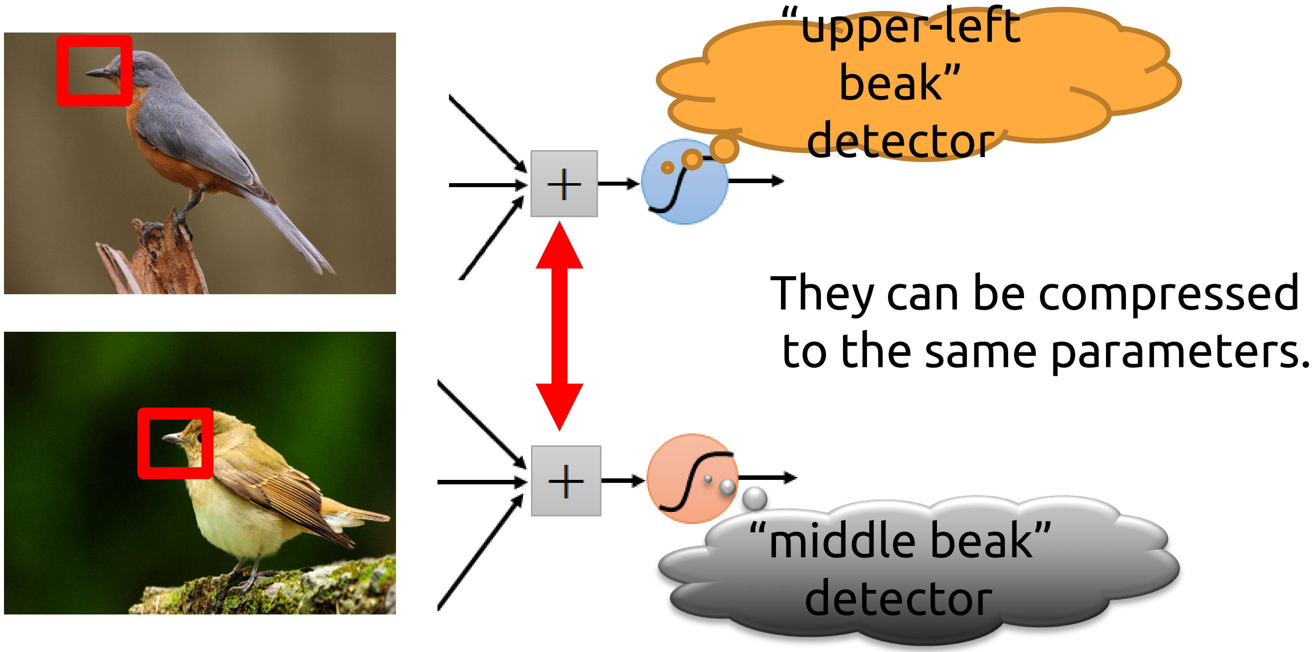
- Some patterns are much smaller than the whole image

Can represent a small region with fewer parameters



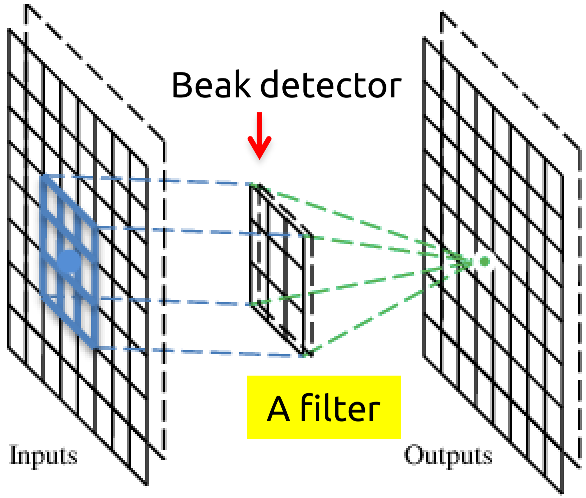
Same pattern appears in different places can be compressed!

What about training a lot of such “small” detectors and each detector must “move around”.



Convolutional Layers

A CNN is a neural network with some convolutional layers (and some other layers). A convolutional layer has a number of filters that does convolutional operation.



https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture5CNN.ppt

Convolution

These are the network parameters to be learned.

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

6 x 6 image

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

⋮ ⋮

Each filter detects a small pattern (3 x 3).

https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture5CNN.ppt

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

stride=1

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

Dot product → 3 -1

6 x 6 image

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

If stride=2

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

6 x 6 image

3 -3

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

stride=1

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

6 x 6 image

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

Convolution

-1	1	-1
-1	1	-1
-1	1	-1

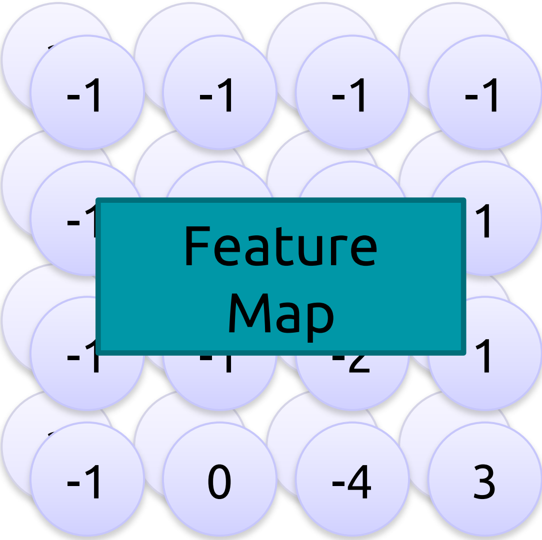
Filter 2

stride=1

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

6 x 6 image

Repeat this for each filter



Two 4 x 4 images Forming 2 x 4 x 4 matrix

https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture5CNN.ppt

Color image: RGB 3 channels

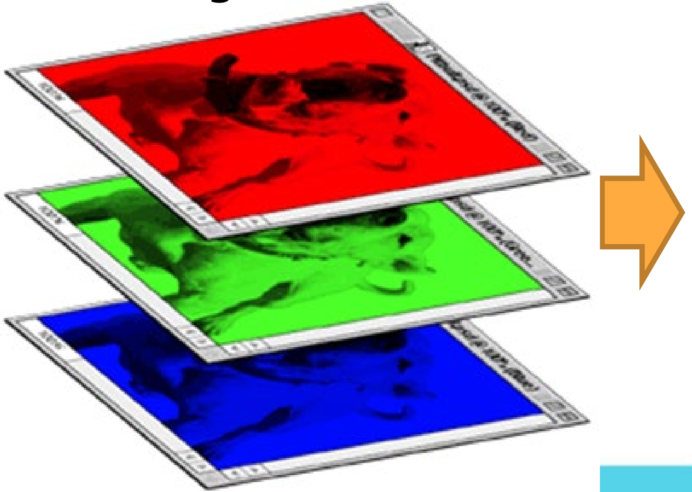
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

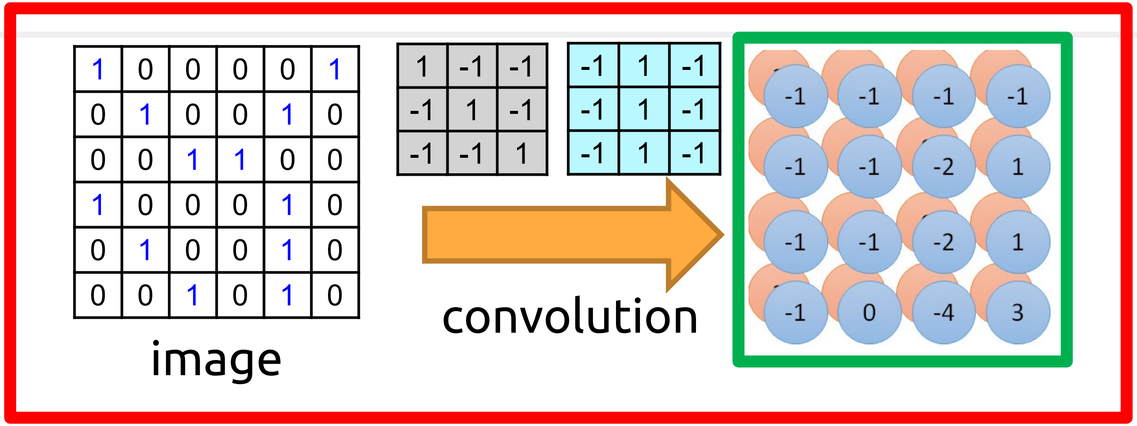
Color image



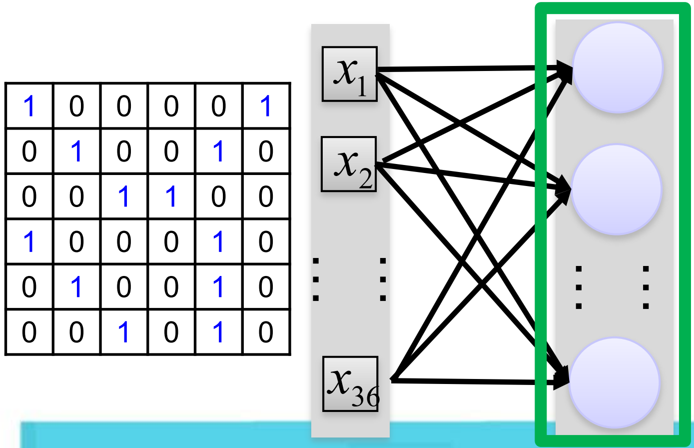
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture5CNN.ppt

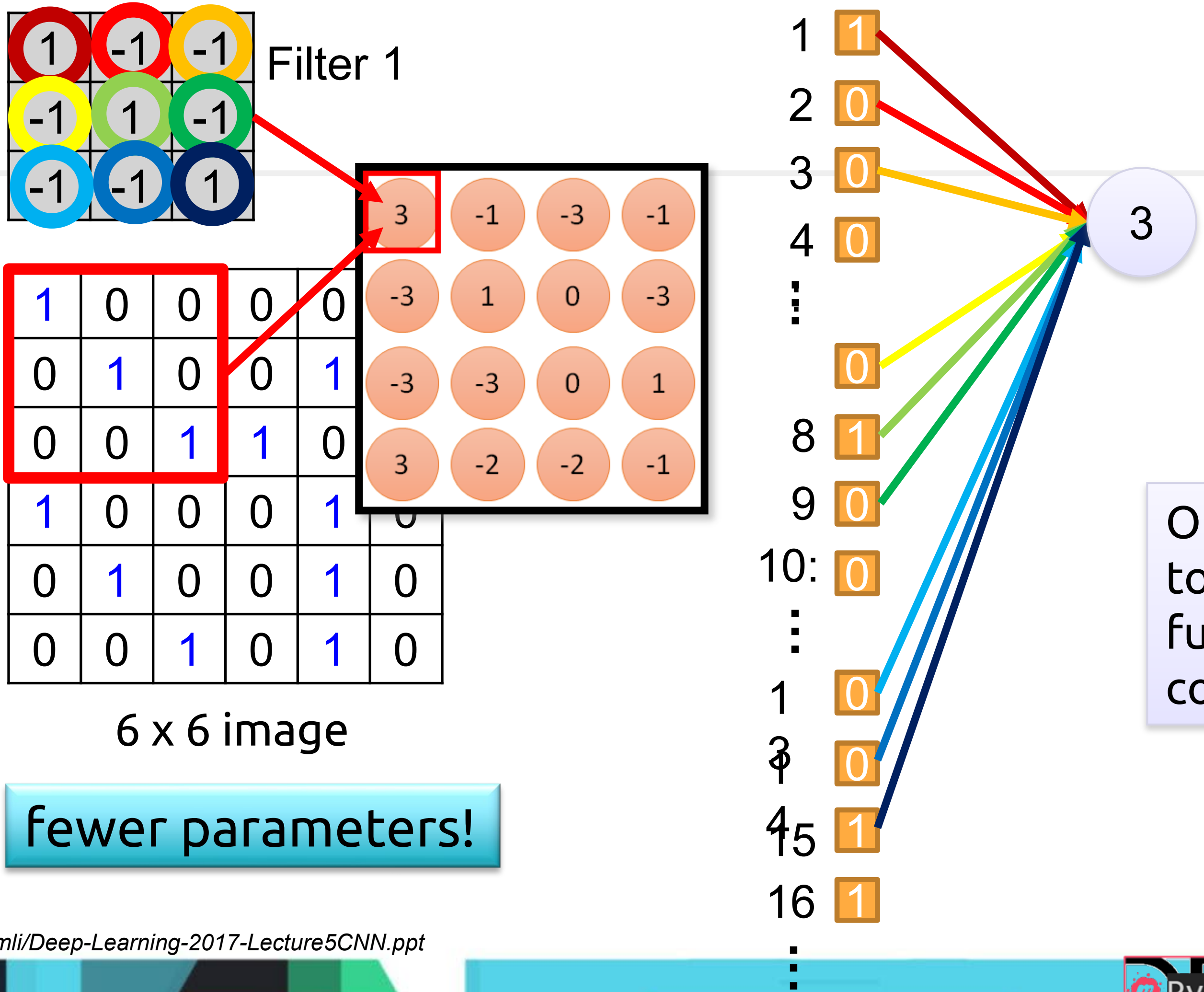
Convolution vs. Fully Connected



Fully-connected



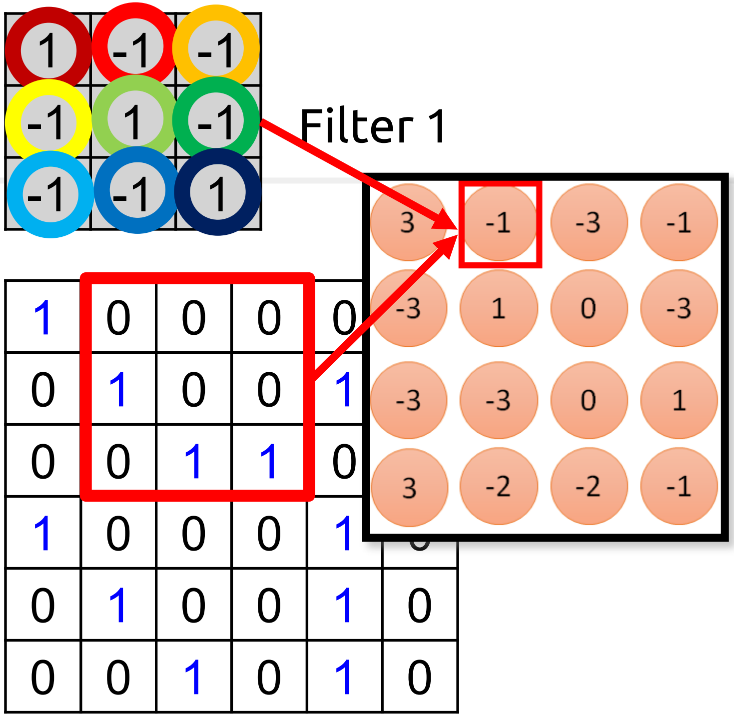
https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture5CNN.ppt



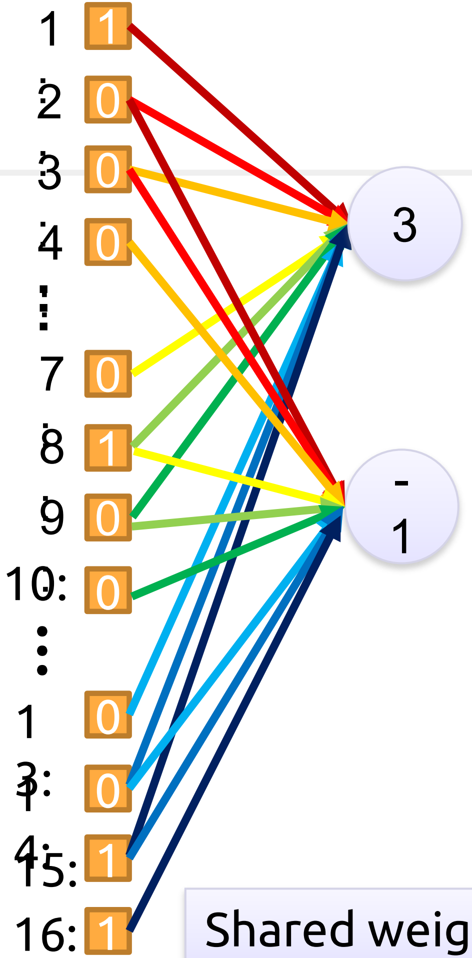
fewer parameters!

Only connect to 9 inputs, not fully connected

https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture5CNN.ppt



6 x 6 image

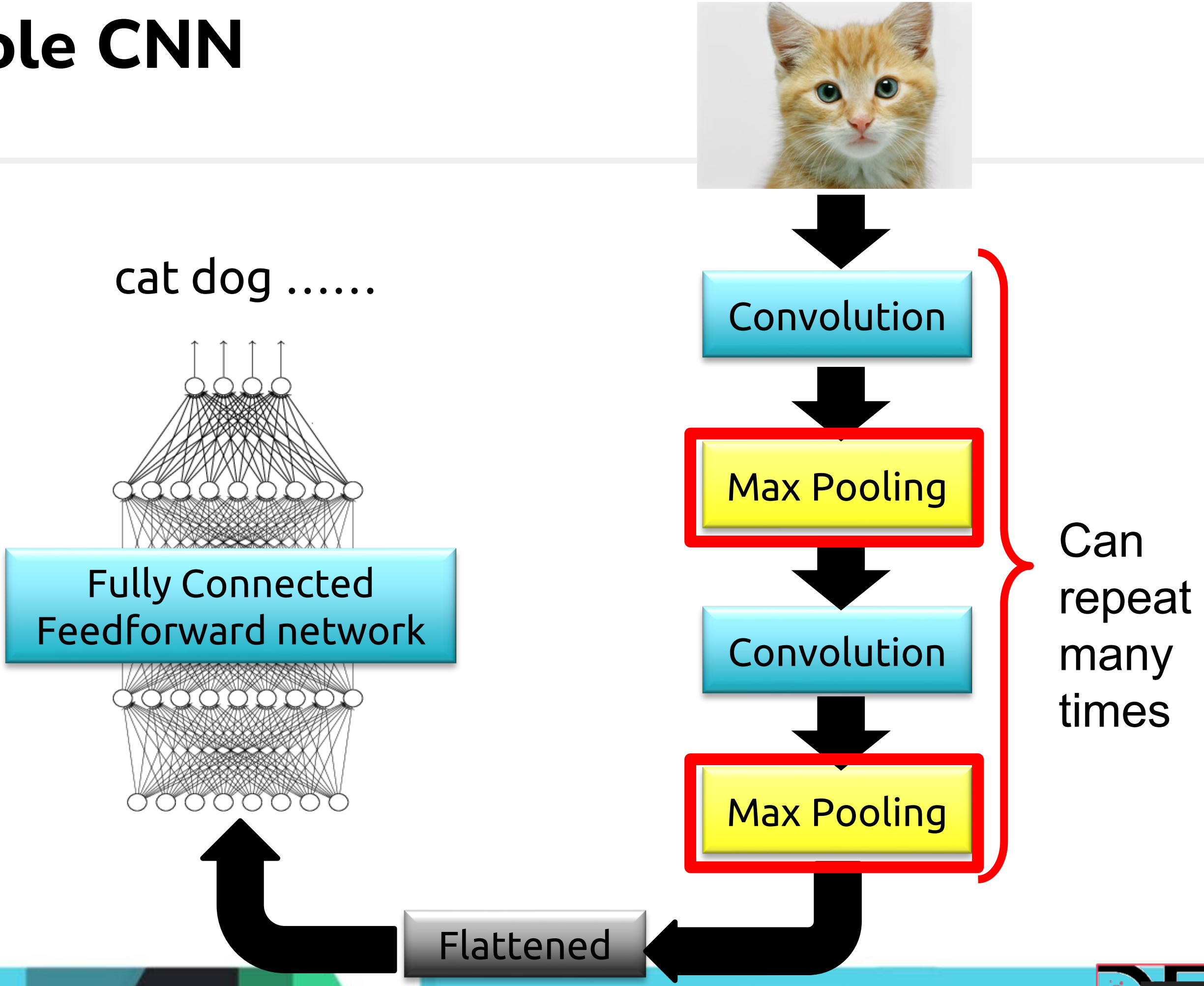


Fewer parameters

Even fewer parameters

Shared weights

The whole CNN



Max Pooling

Filter 1

1	-1	-1
-1	1	-1
-1	-1	1

Filter 2

-1	1	-1
-1	1	-1
-1	1	-1

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

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

Why Pooling

- Subsampling pixels will not change the object

bird

Subsampling

bird

We can subsample the pixels to make image smaller

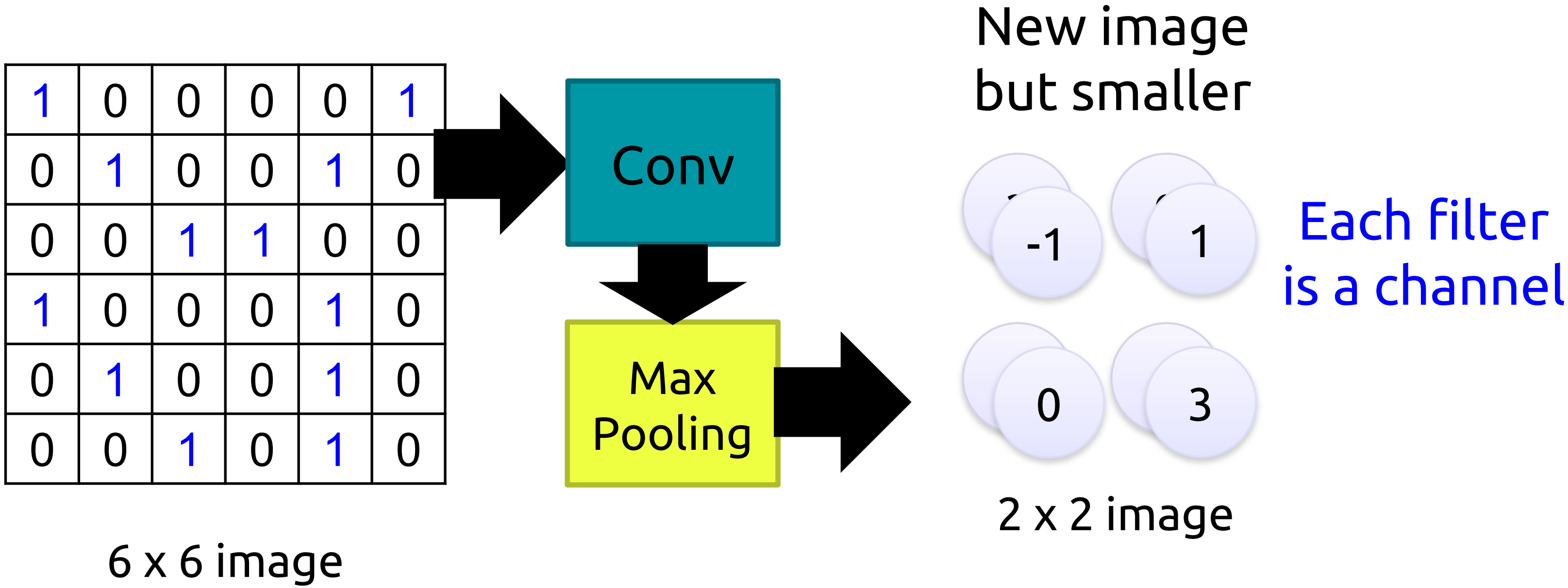
fewer parameters to characterize the image

A CNN compresses a fully connected network in two ways:

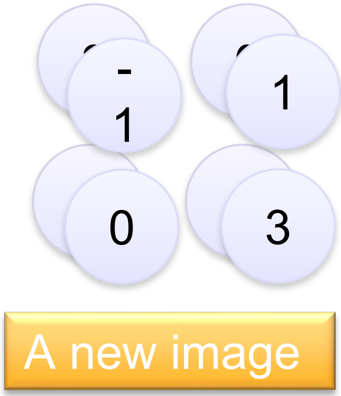
- Reducing number of connections
- Shared weights on the edges
- Max pooling further reduces the complexity

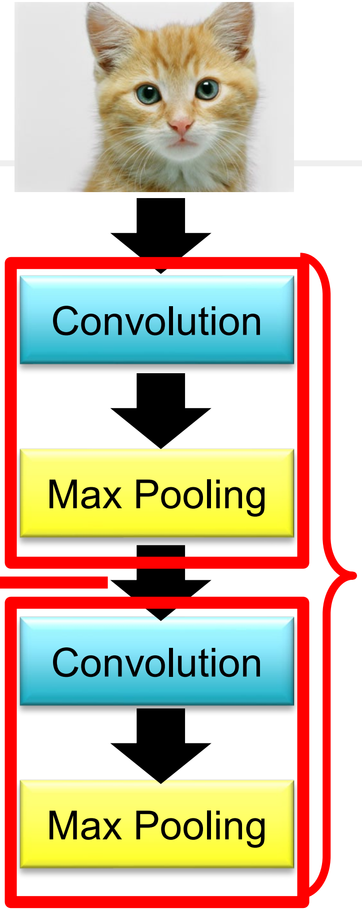
https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture5CNN.ppt

Max Pooling


1 0 0 0 0 1
0 1 0 0 1 0
0 0 1 1 0 0
1 0 0 0 1 0
0 1 0 0 1 0
0 0 1 0 1 0
6 x 6 image
Conv
Max Pooling
New image but smaller
-1
1
0
3
2 x 2 image
Each filter is a channel

The whole CNN



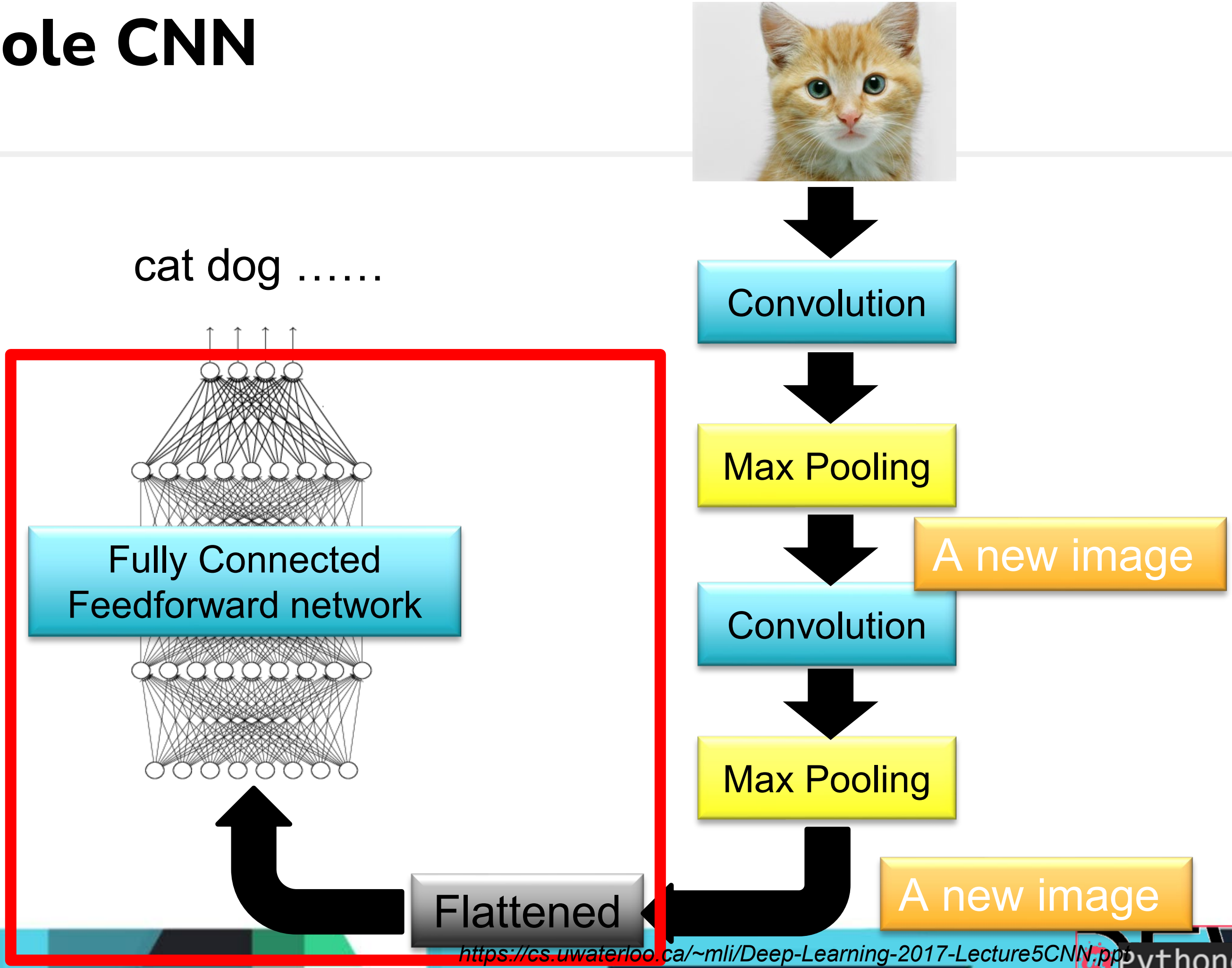


Can repeat many times

Smaller than the original image

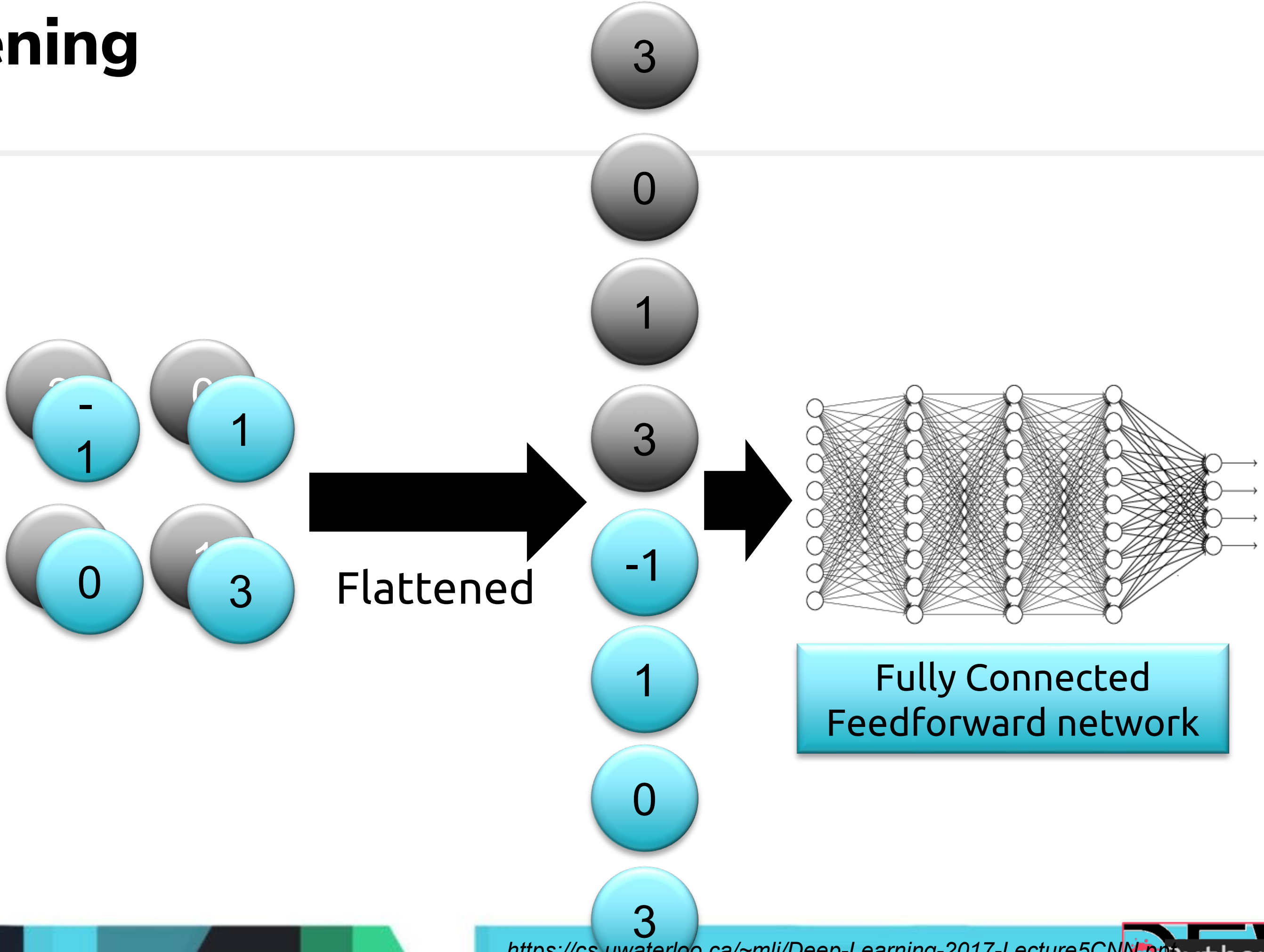
The number of channels is the number of filters

The whole CNN



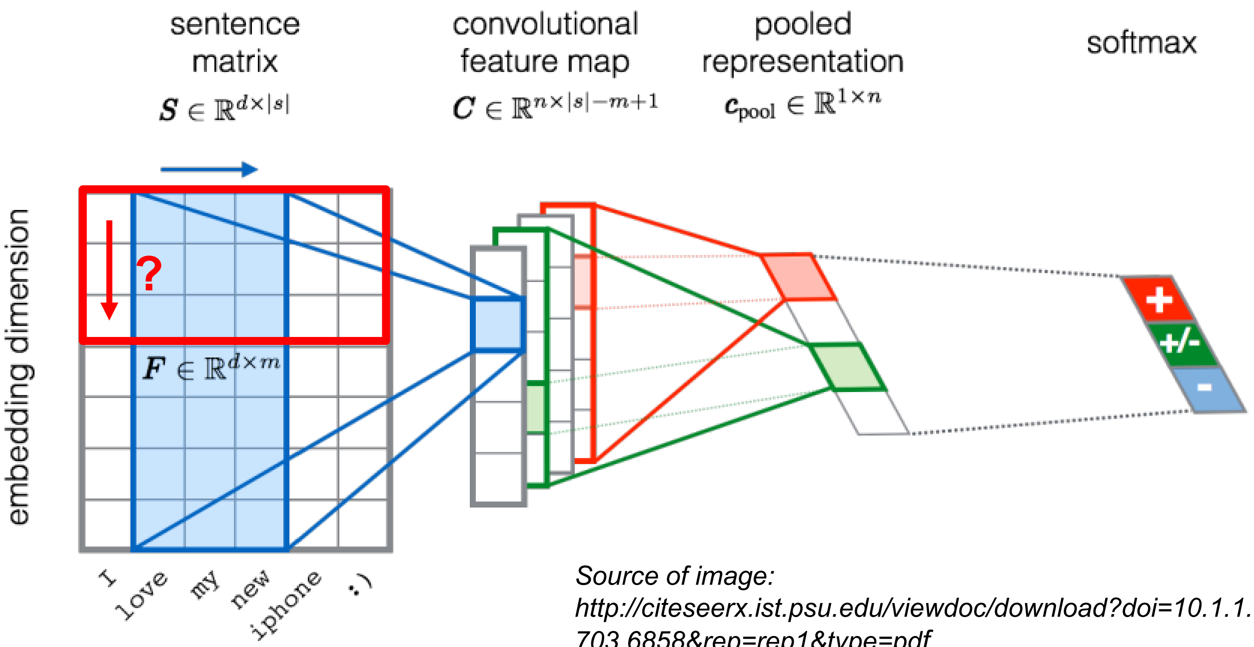
https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture5CNN.ppt

Flattening



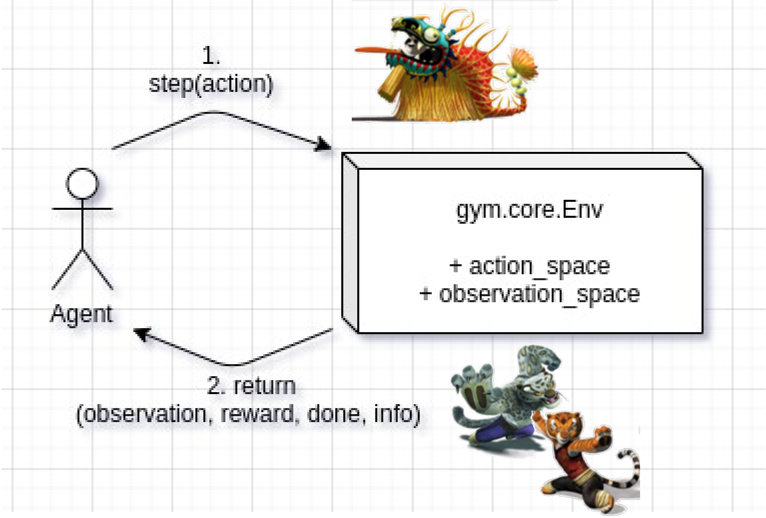
https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture5CNN.ppt

CNN in text classification

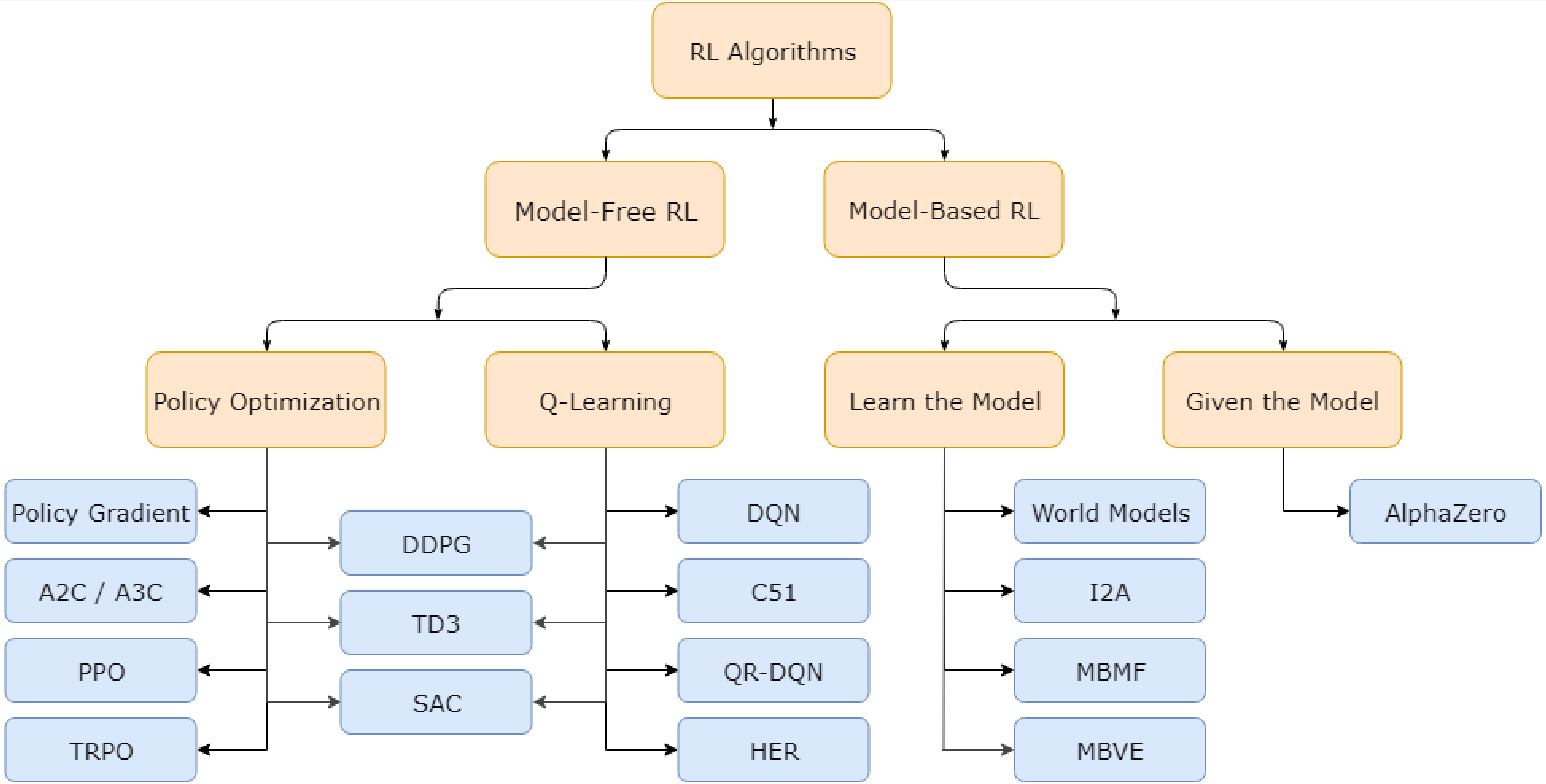


Source of image: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.703.6858&rep=rep1&type=pdf

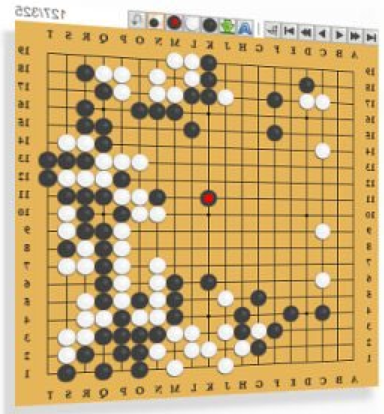
Reinforcement Learning – the Basics



Reinforcement Learning



AlphaGo



19 x 19 matrix

Black: 1

white: -1

none: 0

Neural Network

Next move

(19 x 19 positions)

Fully-connected feedforward network can be used

But CNN performs much better

https://cs.uwaterloo.ca/~mli/Deep-Learning-2017-Lecture5CNN.ppt

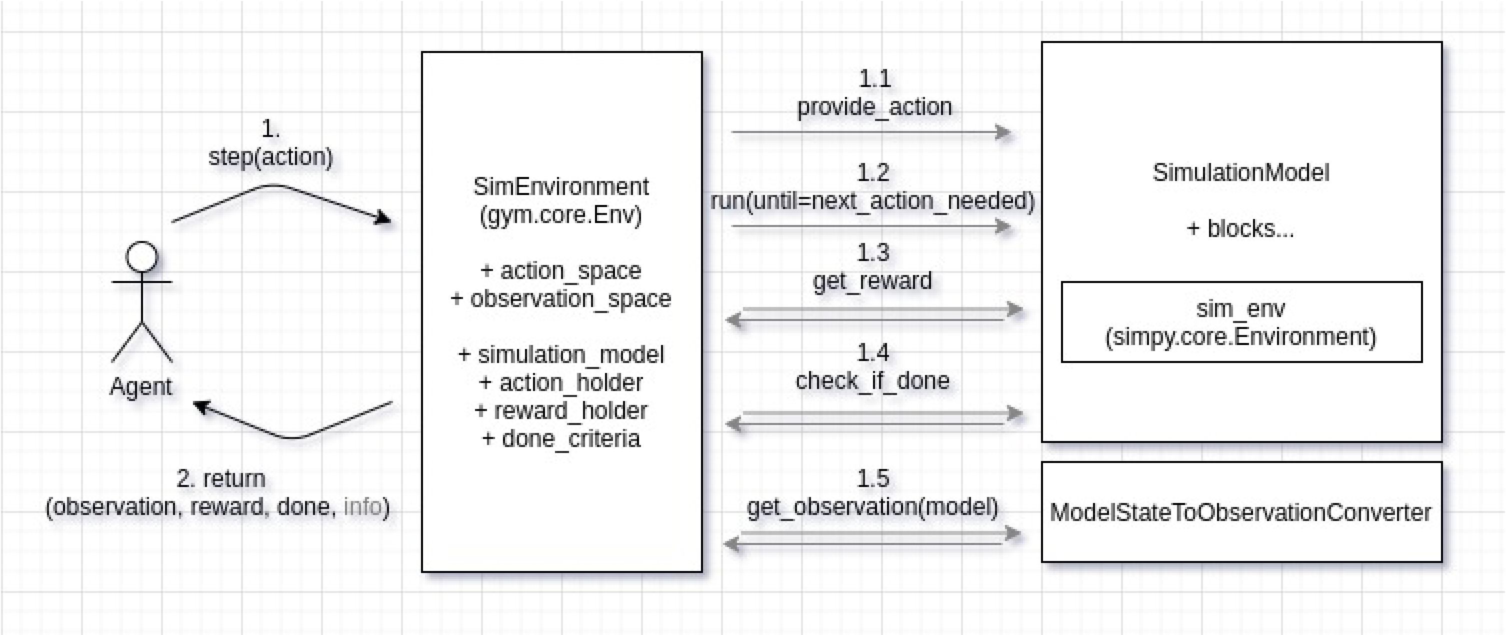
AlphaGo's policy network

The following is quotation from their Nature article:

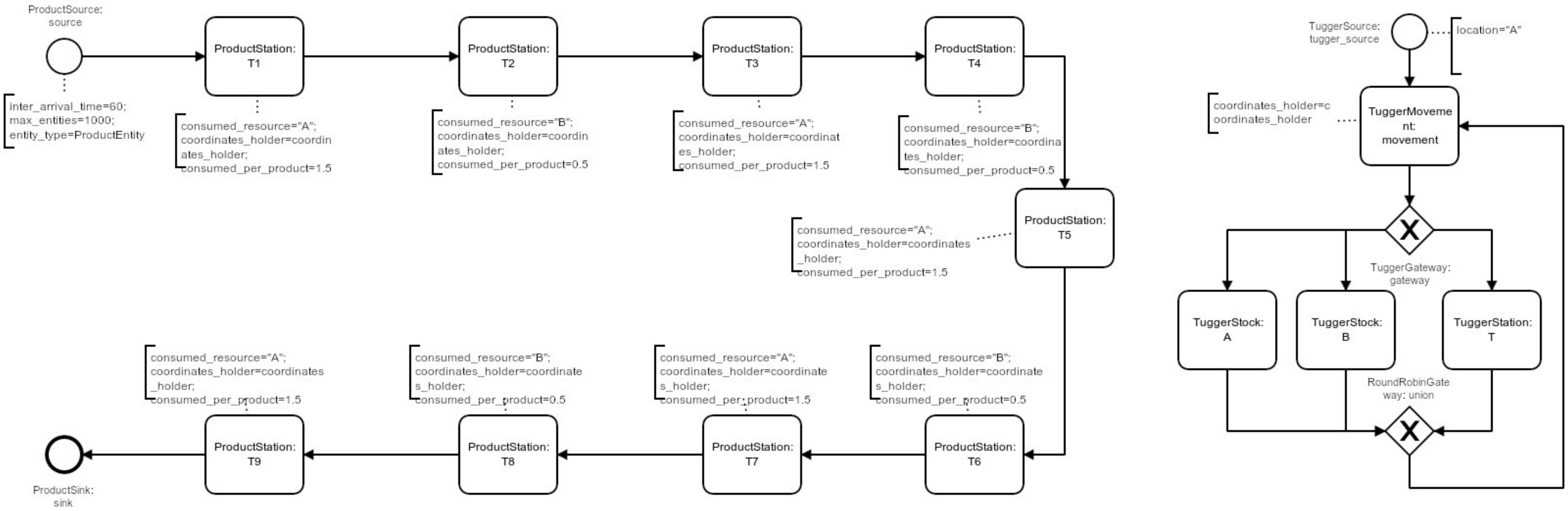
Note: AlphaGo does not use Max Pooling.

Neural network architecture. The input to the policy network is a $19 \times 19 \times 48$ image stack consisting of 48 feature planes. The first hidden layer zero pads the input into a 23×23 image, then convolves k filters of kernel size 5×5 with stride 1 with the input image and applies a rectifier nonlinearity. Each of the subsequent hidden layers 2 to 12 zero pads the respective previous hidden layer into a 21×21 image, then convolves k filters of kernel size 3×3 with stride 1, again followed by a rectifier nonlinearity. The final layer convolves 1 filter of kernel size 1×1 with stride 1, with a different bias for each position, and applies a softmax function. The match version of AlphaGo used $k = 192$ filters; Fig. 2b and Extended Data Table 3 additionally show the results of training with $k = 128$, 256 and 384 filters.

OpenAI's Gym Generalized Use Case

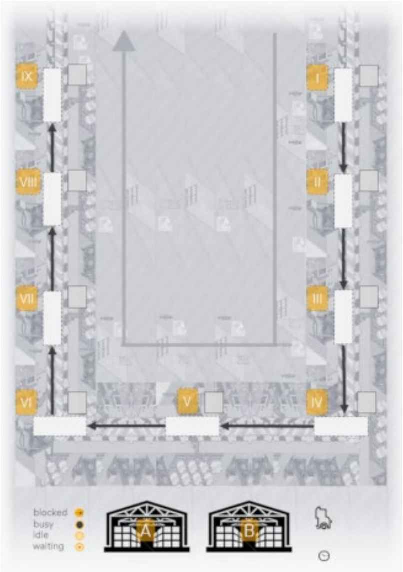


The “gameboard” in Today’s Demo

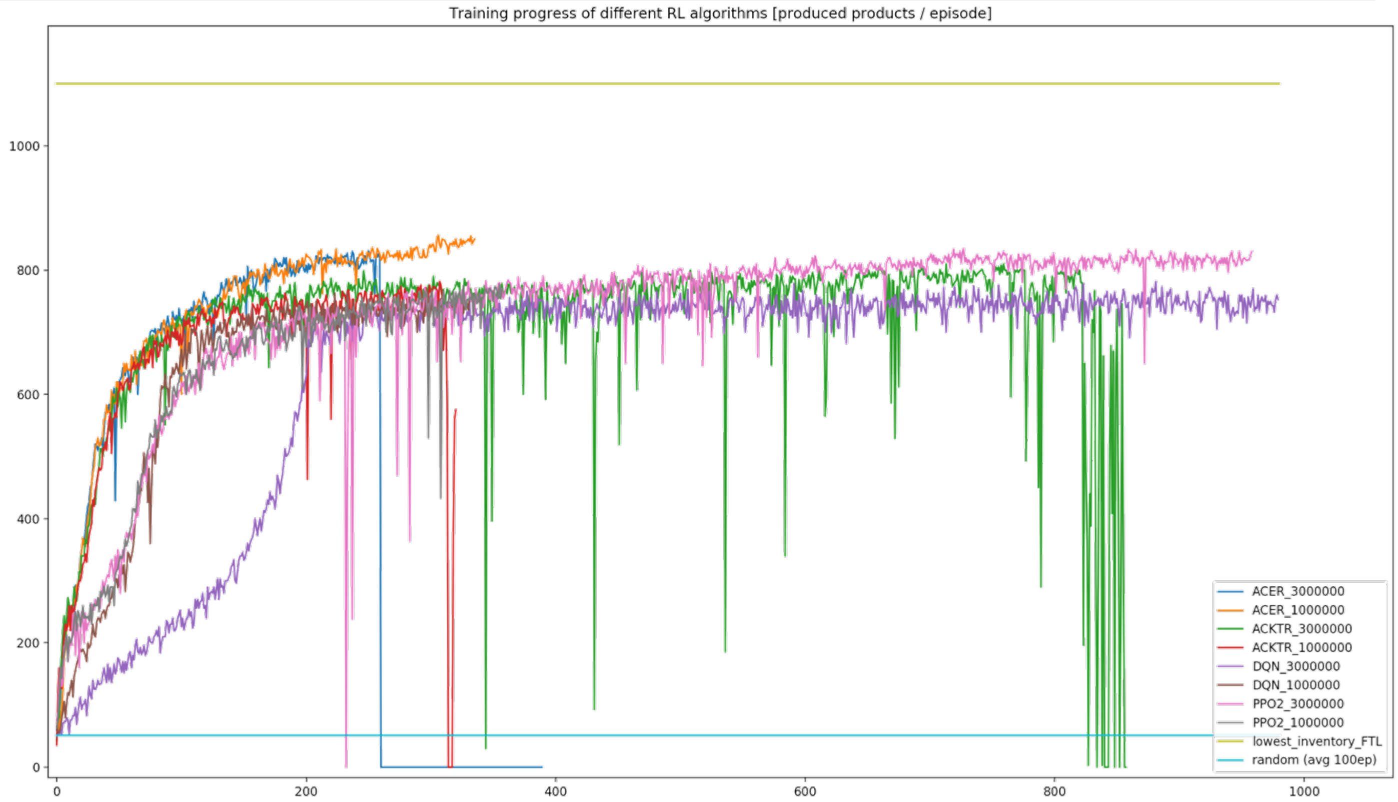


https://fladdimir.github.io/post/tugger-routing/

An Example of PyTorch RL



A “bakeoff” of RL Models



DIVERGENCE ACADEMY

Reinforcement Learning DEMO

"If I have seen further, it is by standing on the shoulders of giants."

-Isaac Newton

Pythoneers

For Future Reference

- http://www.kdnuggets.com/
- http://www.datasciencecentral.com

DIVERGENCE ACADEMY

Questions?

Pythoneers

Thank you!!!

DIVERGENCE ACADEMY
(214) 766-3897
drew.minkin@divergence.one