HANDS ON MACHINE LEARNING

ANEESH@QUILTDATA.IO

@AKARVE

AGENDA

- Theory
- Examples: Deep Neural Network, Regression
- Questions

ARTIFICIAL INTELLIGENCE

- Planning
- Search
- Optimization
- Learning

MACHINE LEARNING IS NOT AI

ML = RATIONAL AGENTS IN A NARROW DOMAINS

- Self-driving cars
- Google Translate
- Amazon Echo, Siri
- AlphaZero
- Watson
- Libratus

AI WINTERS

- 1974-1980
  - Fall of connectionism
  - Perceptrons can’t learn XOR
  - Speech systems underdelivered
- Late 1980s
  - Overhyped LISP machines

SPRING IS COMING

- Powerful hardware
- High-quality, high-volume data
- Limited: algorithmic improvements

MODELS ALWAYS BLOW UP

- Incorrect assumptions
- Unwarranted trust
- Fat tails
- The menu is not the meal

MACHINE LEARNING

- Supervised learning
  - Regression (continuous)
  - Classification (discrete)
- Unsupervised learning
  - Clustering
  - Dimensionality reduction
  - Density estimation

CRASH COURSE: NEURAL NETWORKS

ITS TENSORS ALL THE WAY DOWN

- Rank 0: Scalar = c
- Rank 1: Vector = [v1, v2, v3…]
- Rank 2: Matrix = M x N
- Rank 3+: Tensor = M x N x O x …

FEATURES, PARAMETERS, CLASSIFICATION

$$\underset{\textit{A (Features)}}{\begin{bmatrix} 0.3 & 0.4 & 0.2 \\ 0.1 & 0.2 & -0.1 \\ 0.1 & 0.2 & -0.1 \end{bmatrix}} \underset{\textit{x (Parameters)}}{\begin{bmatrix} p1 \\ p2 \\ p3 \end{bmatrix}} = \underset{\textit{b (Classes)}}{\begin{bmatrix} \text{“up”} \\ \text{“up”} \\ \text{“down”} \end{bmatrix}}$$

FEATURES, PARAMETERS, CLASSIFICATION

A = *Features* *x* = *Parameters* *b* = *Classes*

$$\begin{bmatrix} 0.3 & 0.4 & 0.2 \\ \boxed{0.1 \quad 0.2 \quad -0.1 \quad \text{TRAINING VECTOR}} \\ 0.1 & 0.2 & -0.1 \end{bmatrix} \begin{bmatrix} p1 \\ p2 \\ p3 \end{bmatrix} = \begin{bmatrix} \text{“up”} \\ \text{“up”} \\ \text{“down”} \end{bmatrix}$$

FEATURES, PARAMETERS, CLASSIFICATION

A = *Features* x = *Parameters* b = *Classes*

$$\begin{bmatrix} 0.3 & 0.4 & 0.2 \\ 0.1 & 0.2 & -0.1 \\ 0.1 & 0.2 & -0.1 \end{bmatrix} \begin{bmatrix} p1 \\ p2 \\ p3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$$

TRAINING VECTOR

FEATURES, PARAMETERS, CLASSIFICATION

A = Features *x = Parameters* *b = Classes*

$$\begin{bmatrix} 0.3 & 0.4 & 0.2 \\ 0.1 & 0.2 & -0.1 \\ 0.1 & 0.2 & -0.1 \end{bmatrix} \begin{bmatrix} 0.1 \\ -0.1 \\ 0.1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \quad -.002$$

TRAINING VECTOR

LOSS

PERCEPTRON [DLPA17]

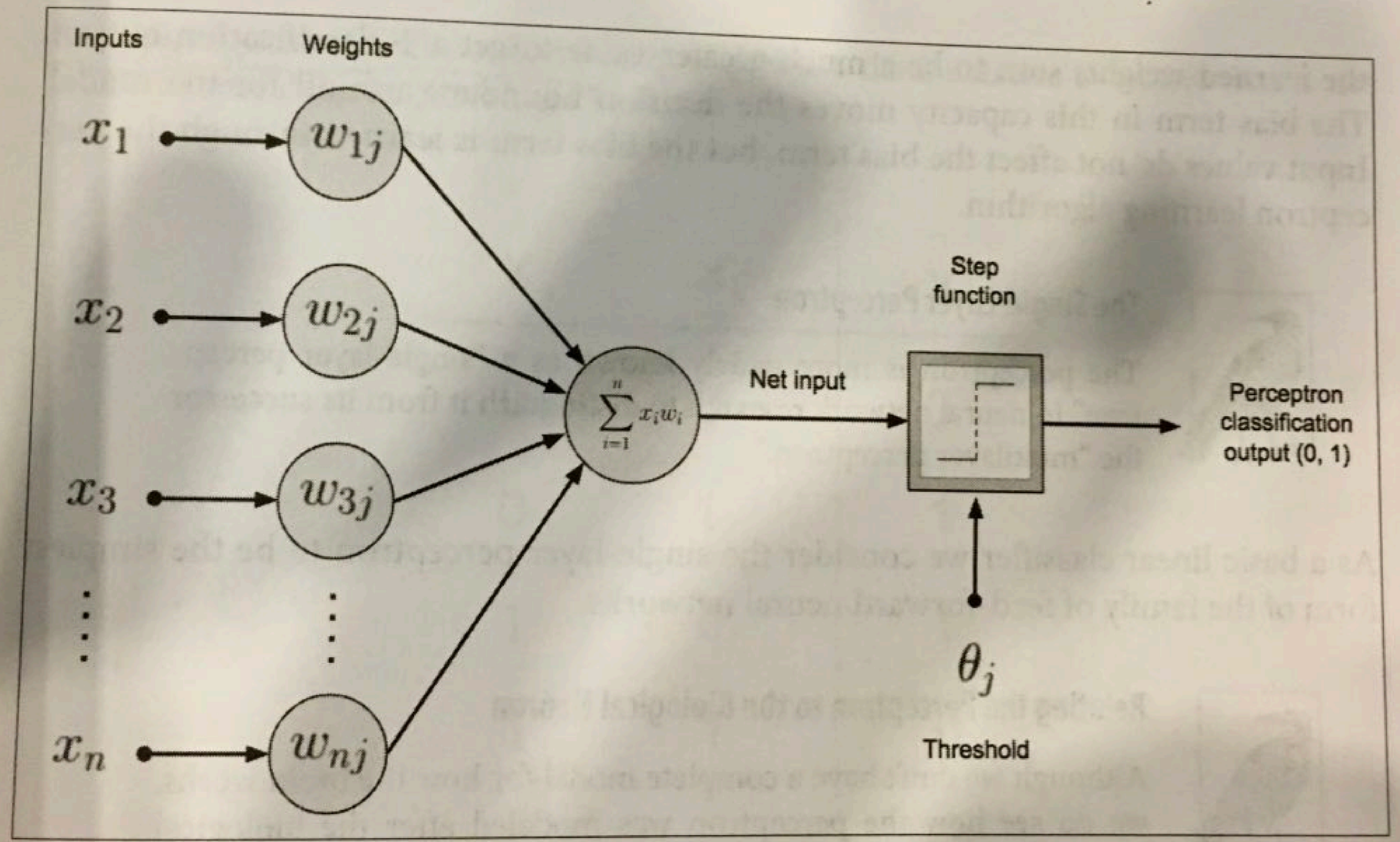


Figure 2-3. Single-layer perceptron

NN TOPOLOGY [DLPA17]

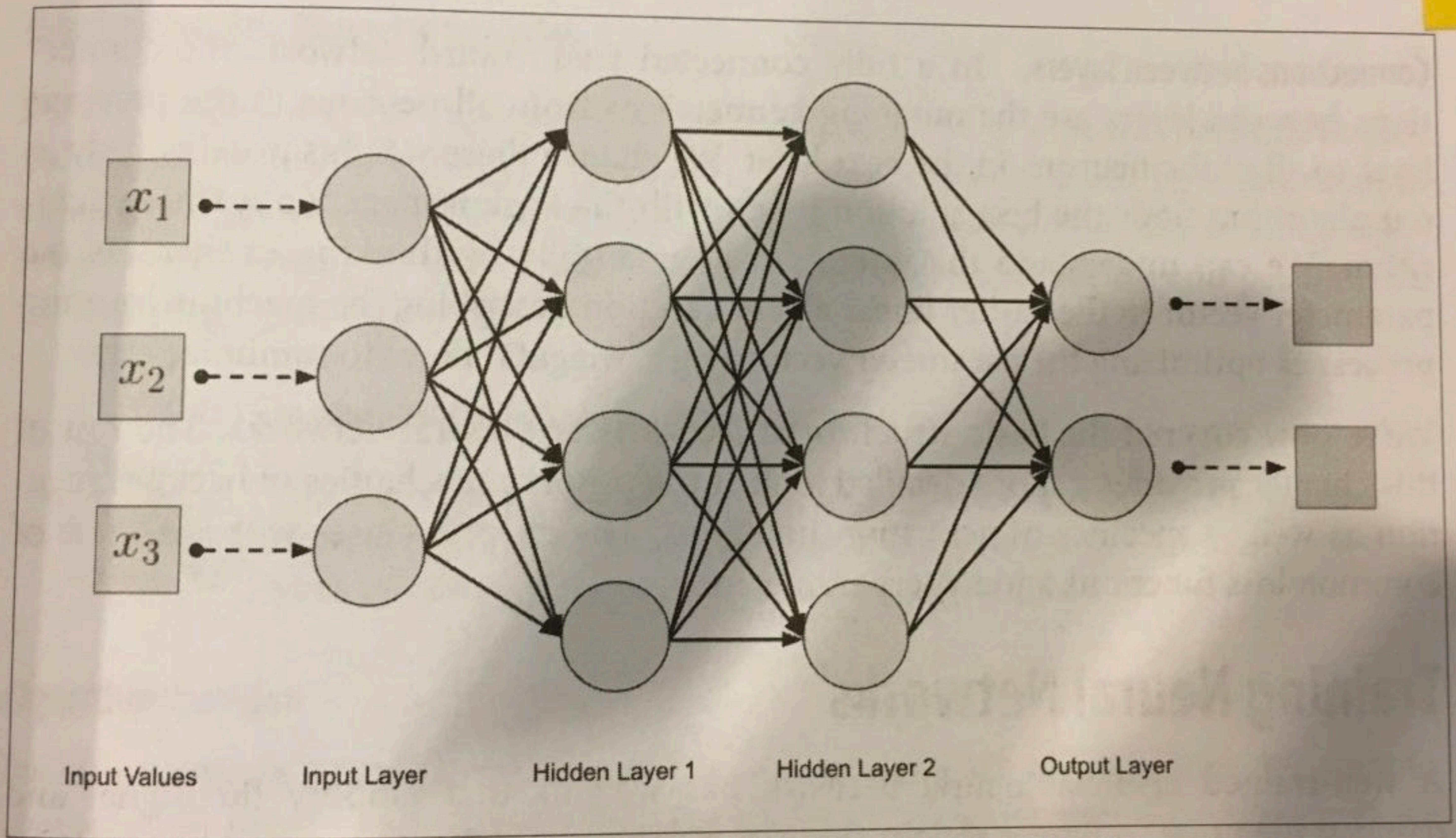
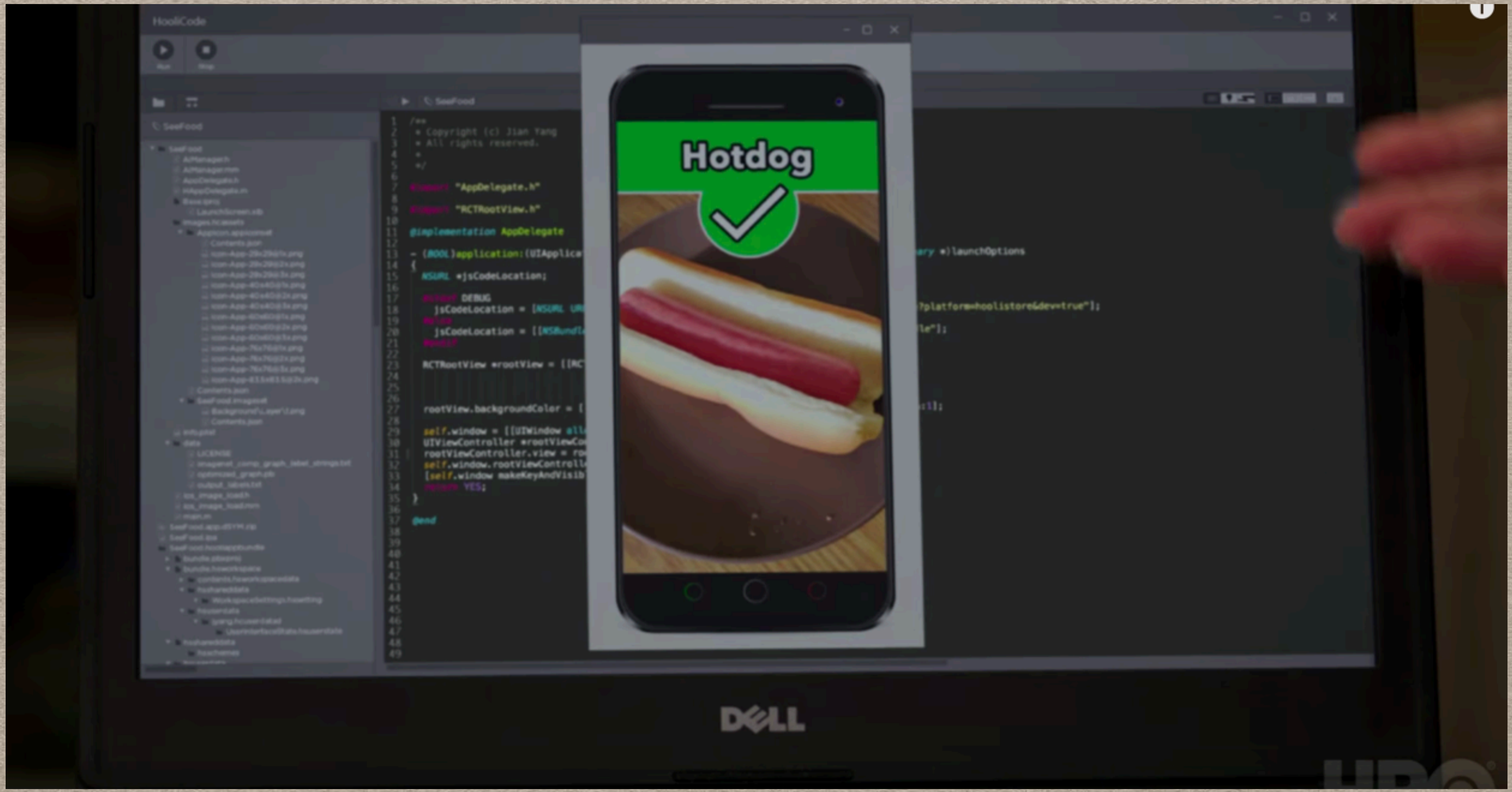


Figure 2-7. Fully connected multilayer feed-forward neural network topology



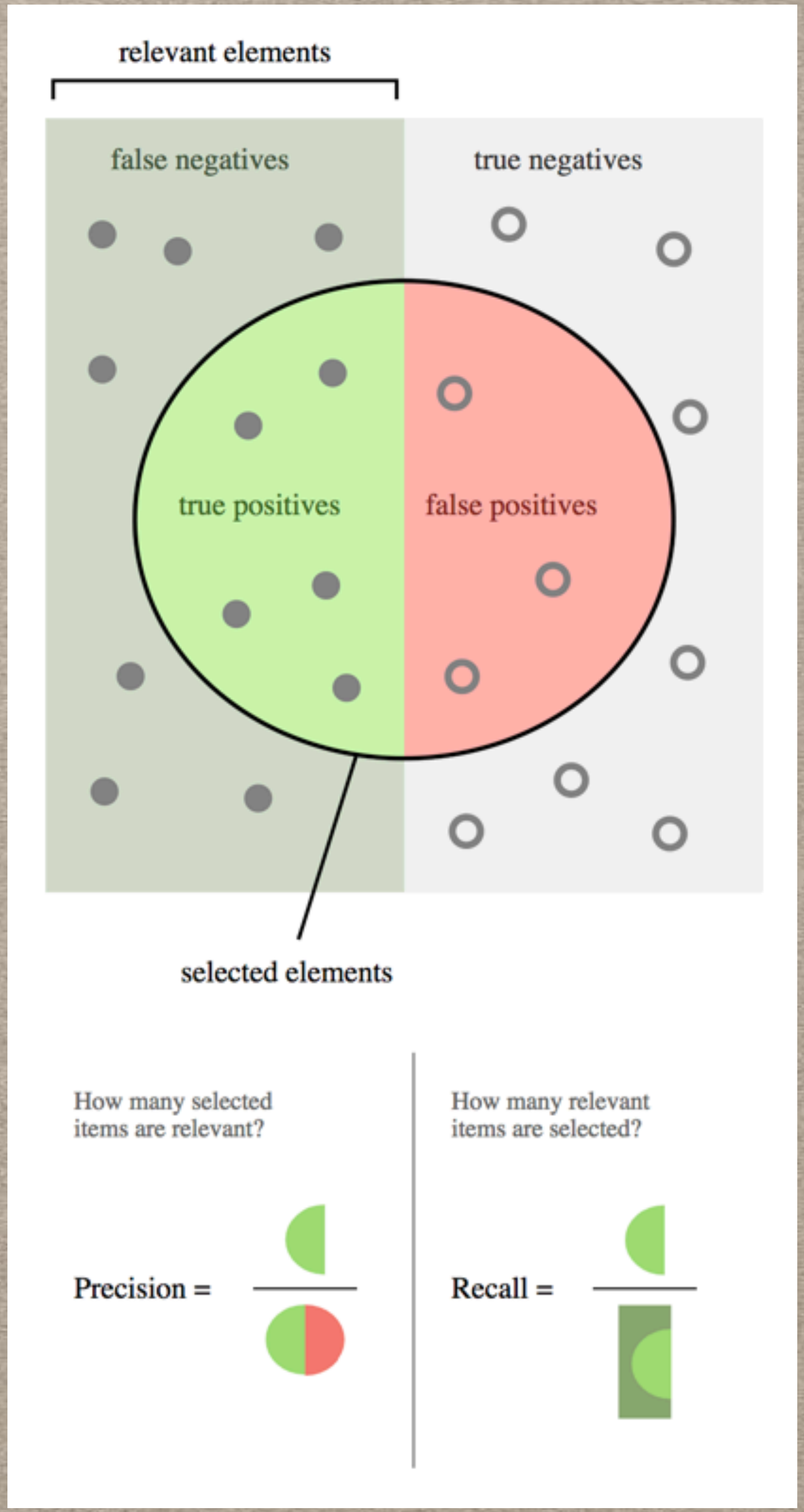
https://www.youtube.com/watch?v=ACmydtFDTGs

MEASURING ACCURACY

- True Positive (TP) = 1 predicted, 1 actual
- True Negative (TN) = 0 predicted, 0 actual
- False Positive (FP) = 1 predicted, 0 actual
- False Negative = 0 predicted, 1 actual

MEASURING ACCURACY

- Precision = TP/(TP + FP)
- Recall = TP/(TP + FN)
- F1-Score
- Confusion matrix

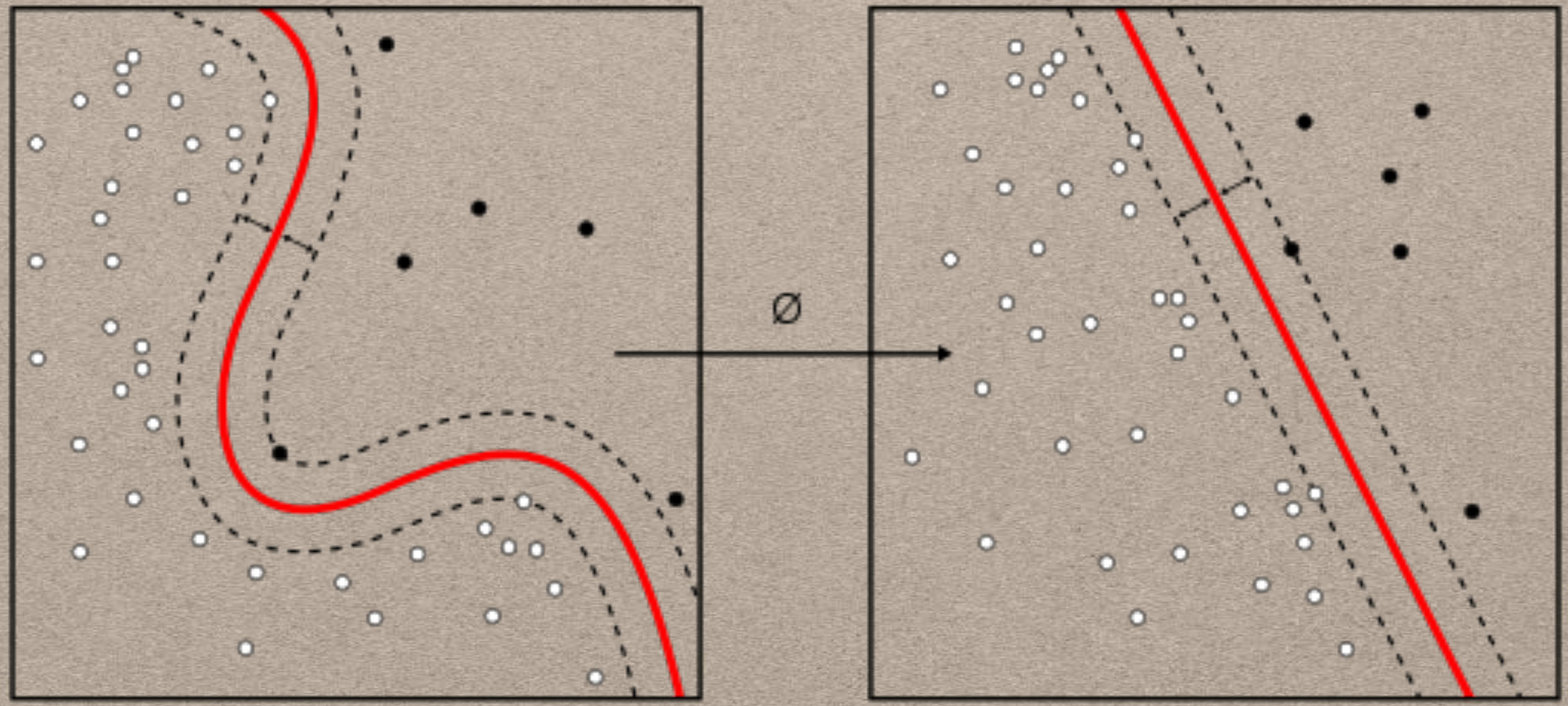


EXAMPLE I: TIME SERIES PREDICTION WITH TENSOR FLOW

TRAINING VS INFERENCE

- Training - long, computationally expensive
- Inference - fast, cheap

TRAINING LEARNS A DECISION SURFACE IN FEATURE SPACE



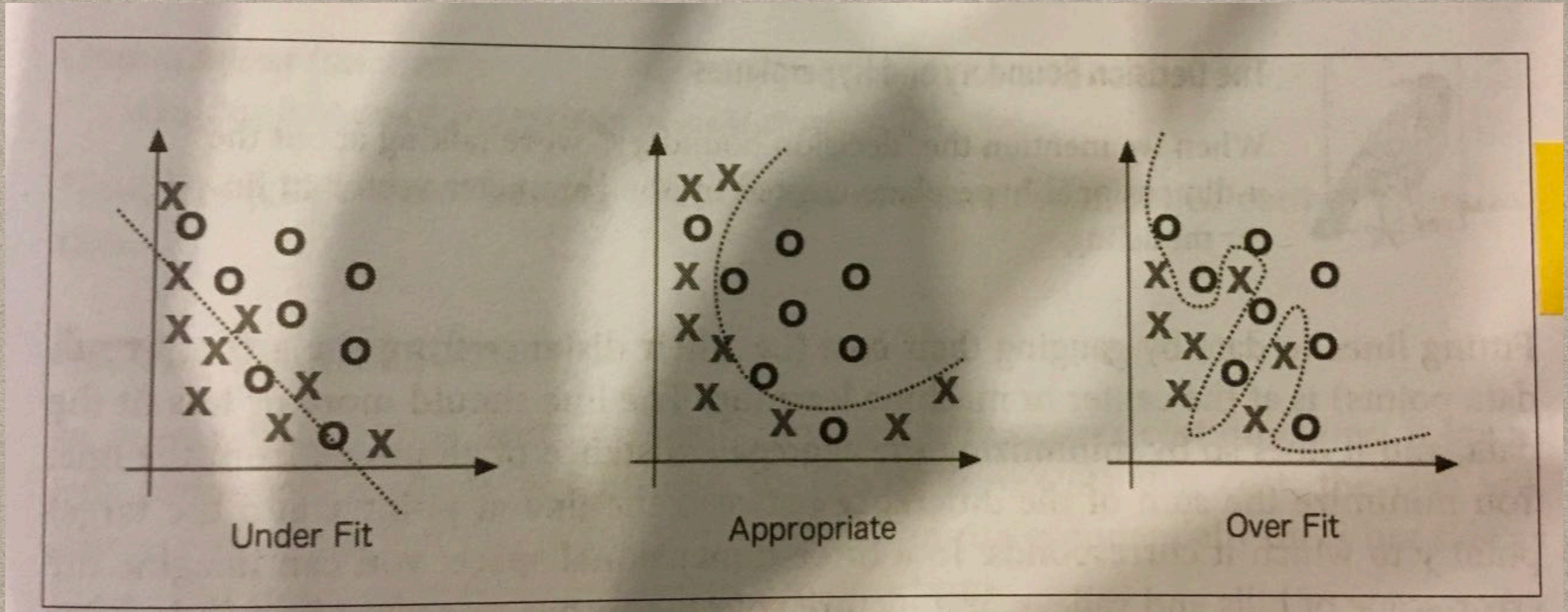


Figure 1-7. Underfitting and overfitting in machine learning

OVERFITTING

INABILITY TO GENERALIZE

AVOIDING OVERFITTING

- Simplest possible models
- Validation
- Regularization
- Dropout
- Data augmentation (noise, pre-processing, etc.)

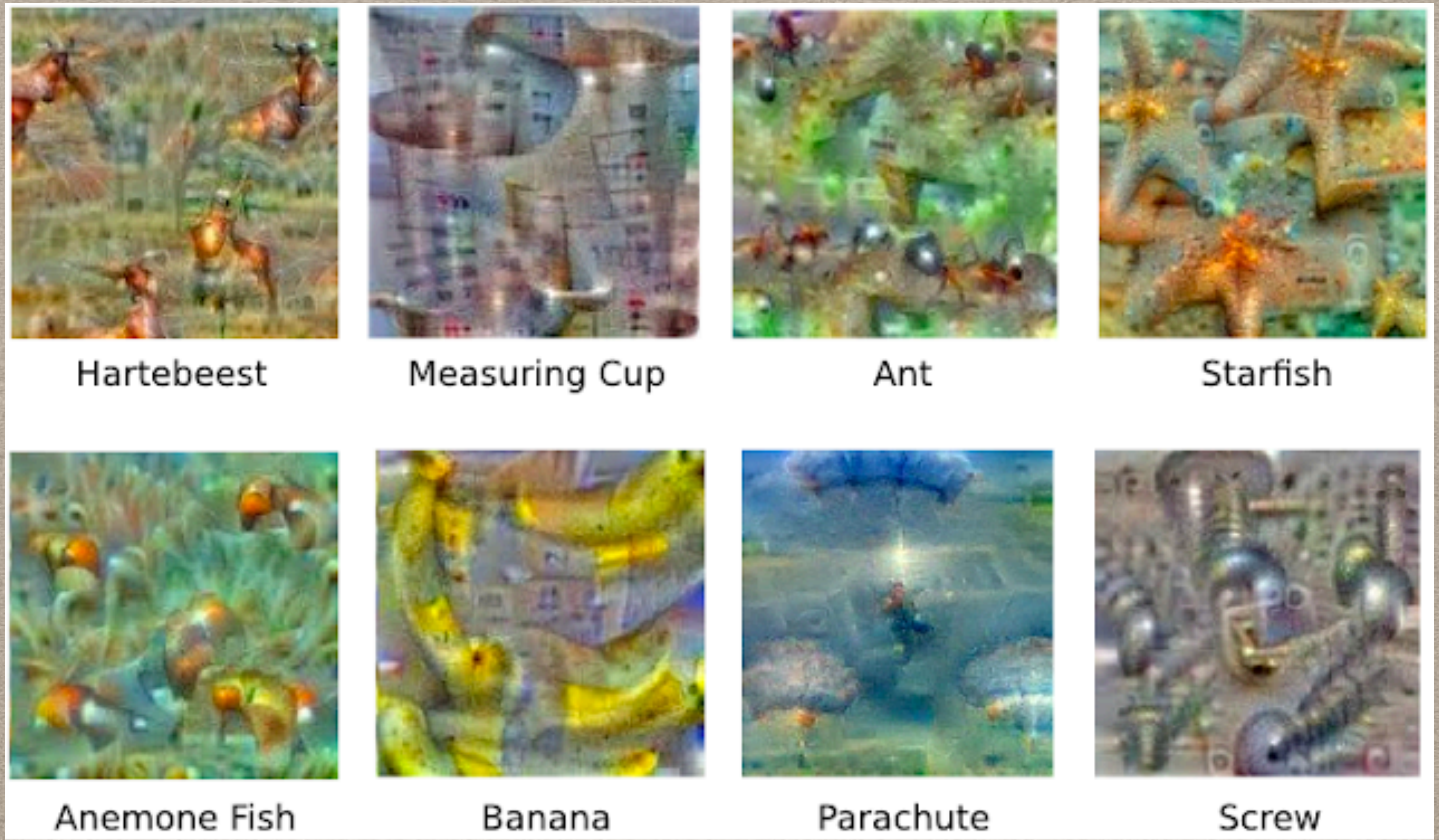
THREE LAWS OF ML

- Inductive bias
- No free lunch (NFL)
- Curse of dimensionality

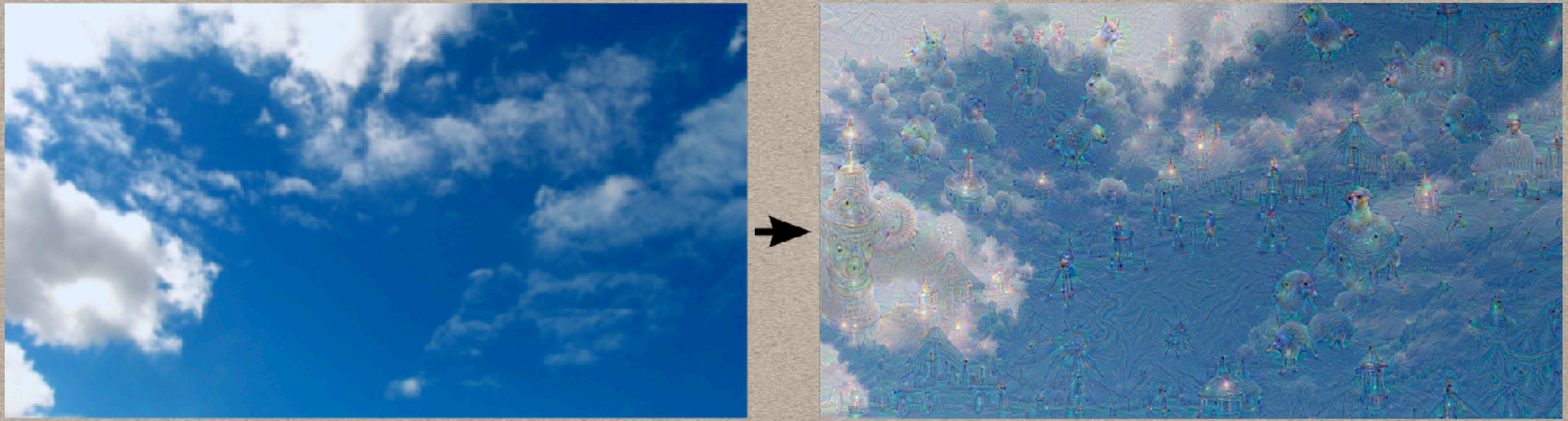
INDUCTIVE BIAS

- Inductive bias = what the algorithm infers that is not in the training data
- Rote learner cannot generalize (can only memorize)
- Each classifier has its own inductive bias

INCEPTIONISM

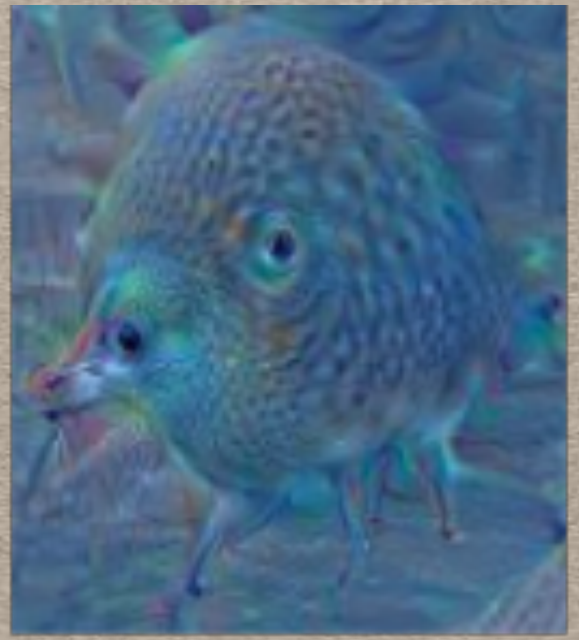


https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html



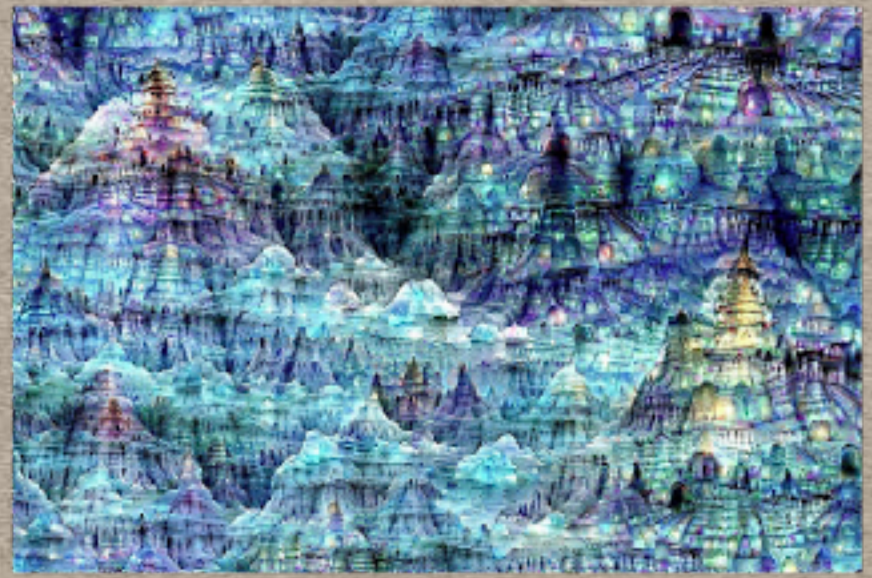
"Admiral Dog!"

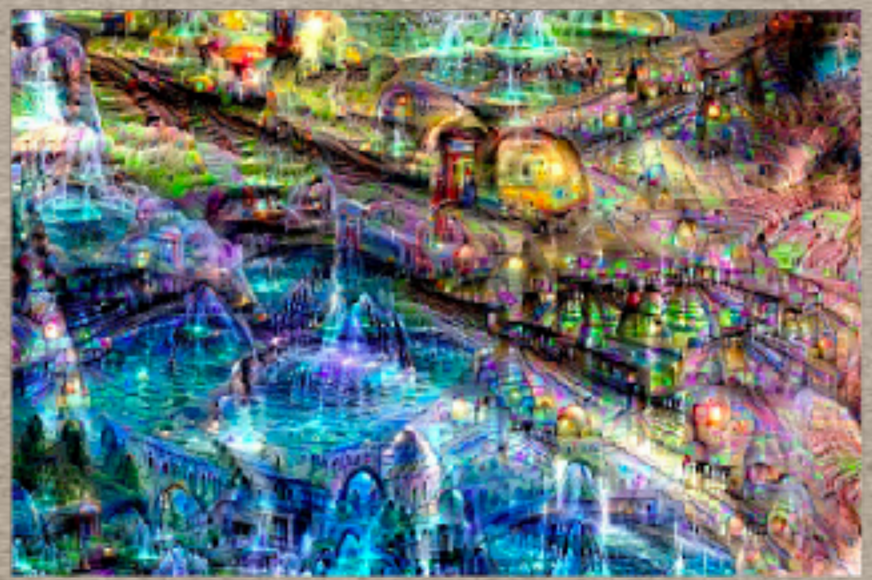
"The Pig-Snail"

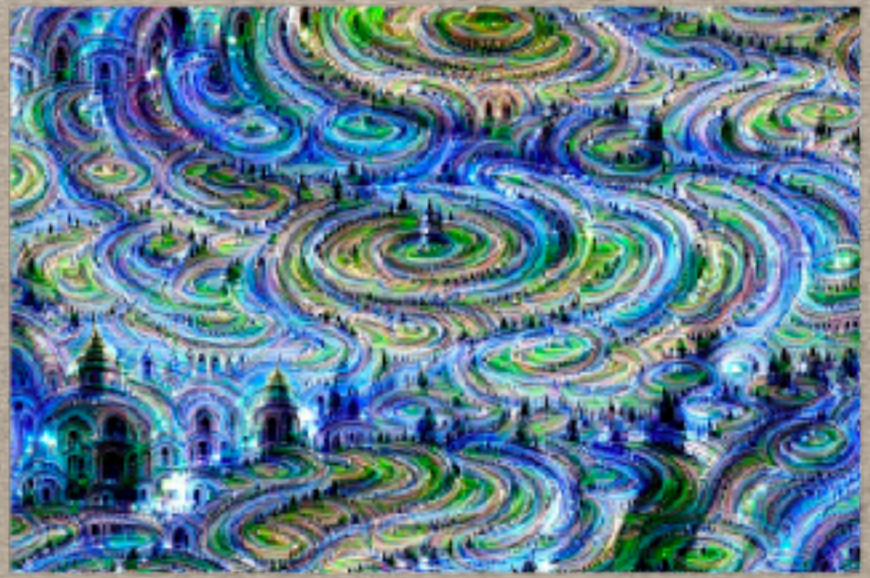


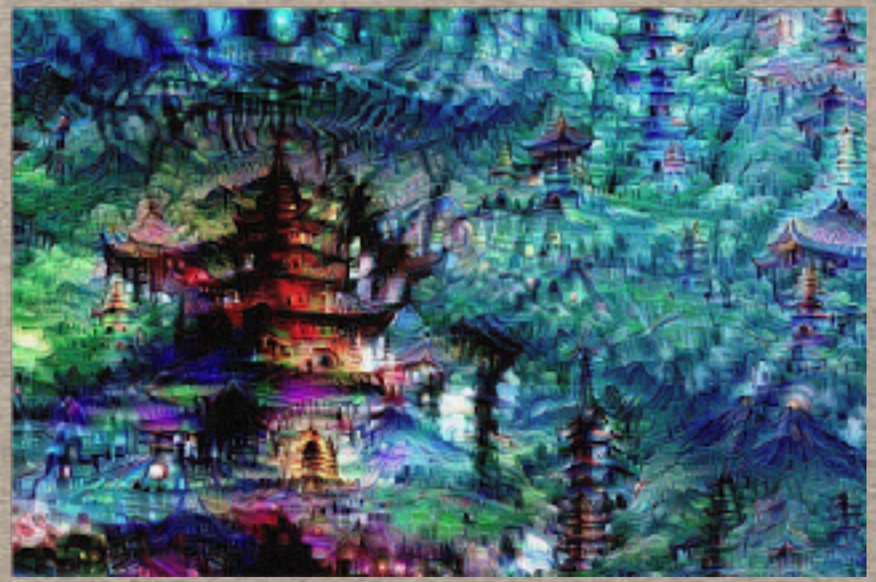
"The Camel-Bird"

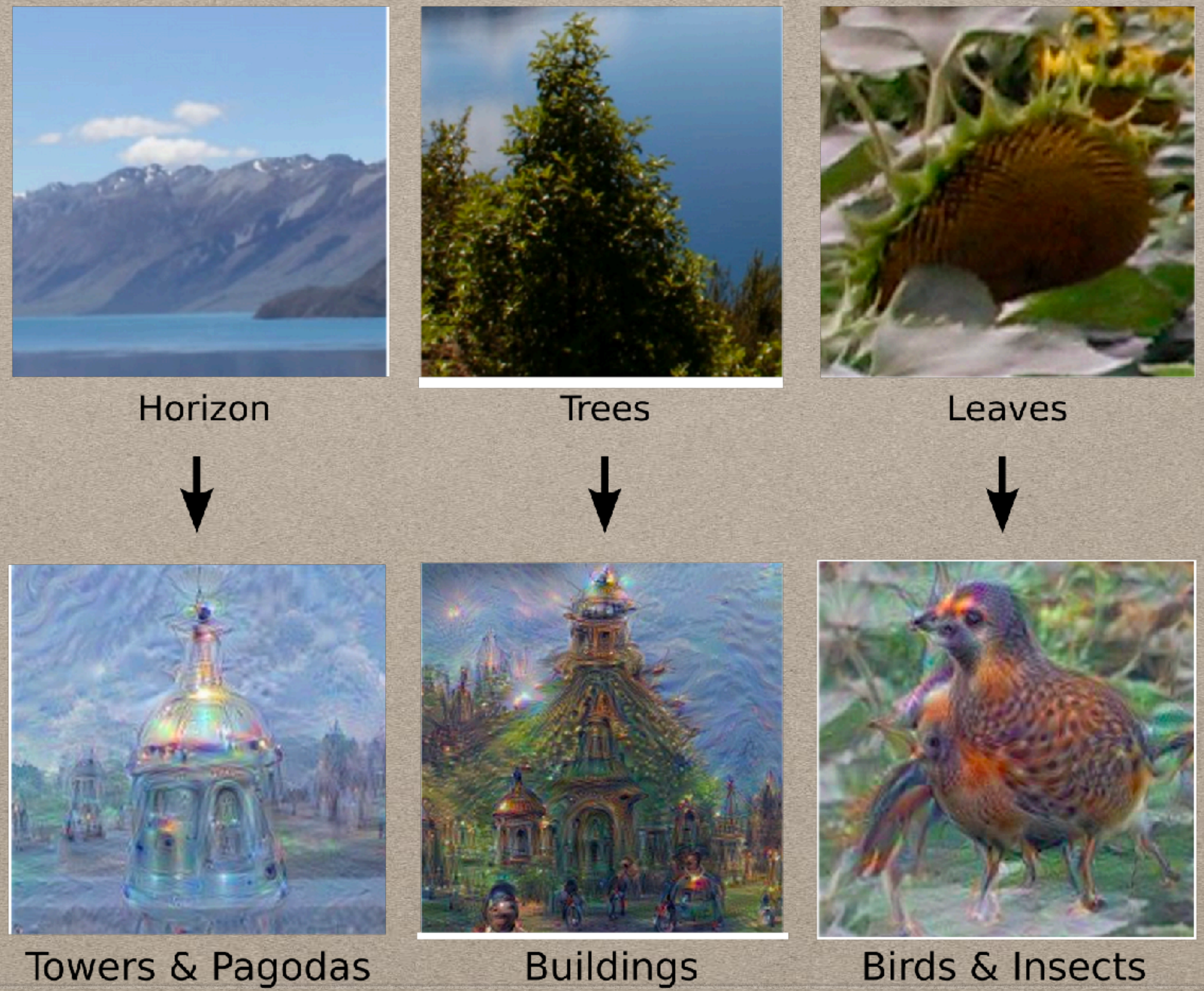
"The Dog-Fish"





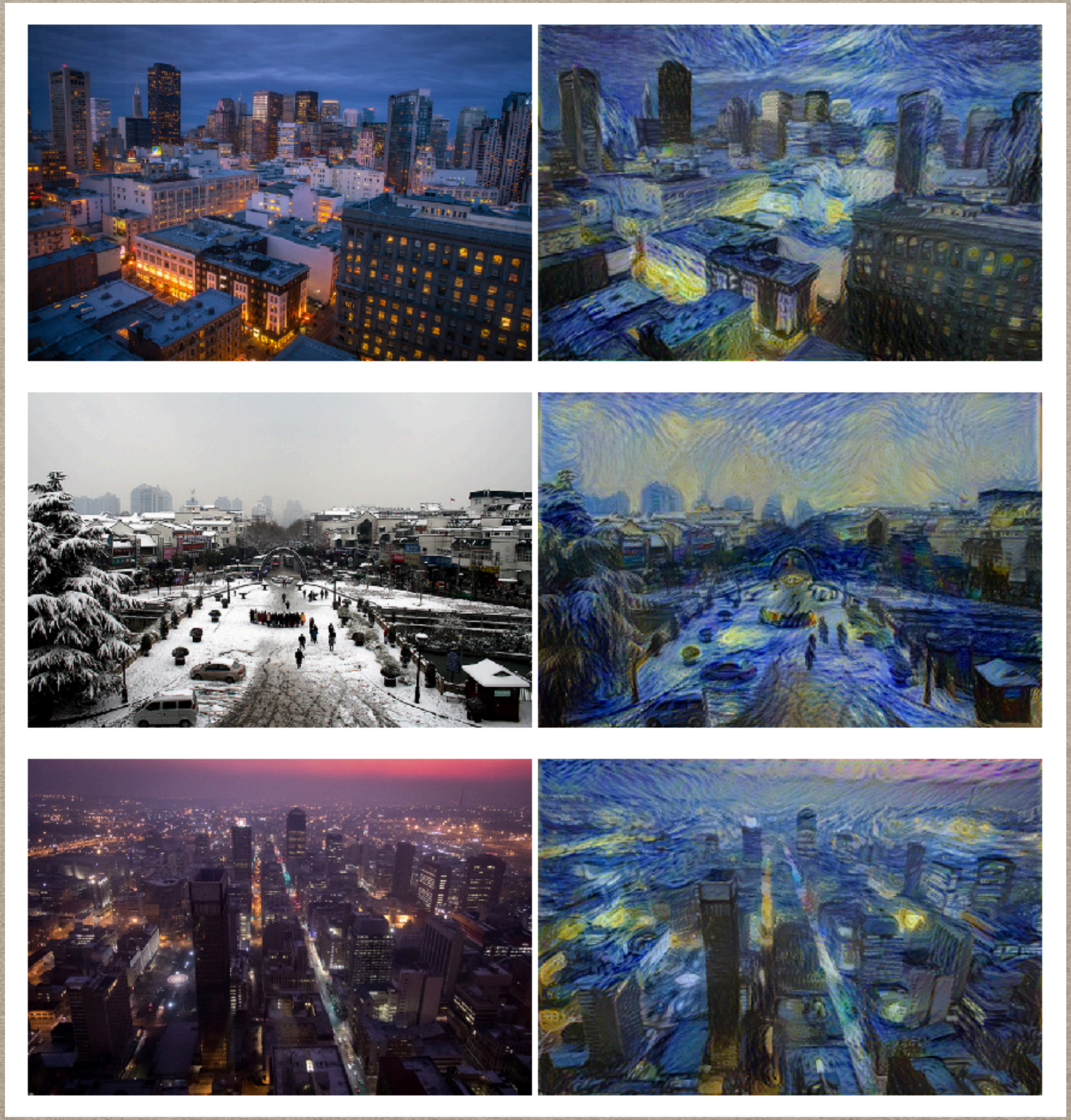


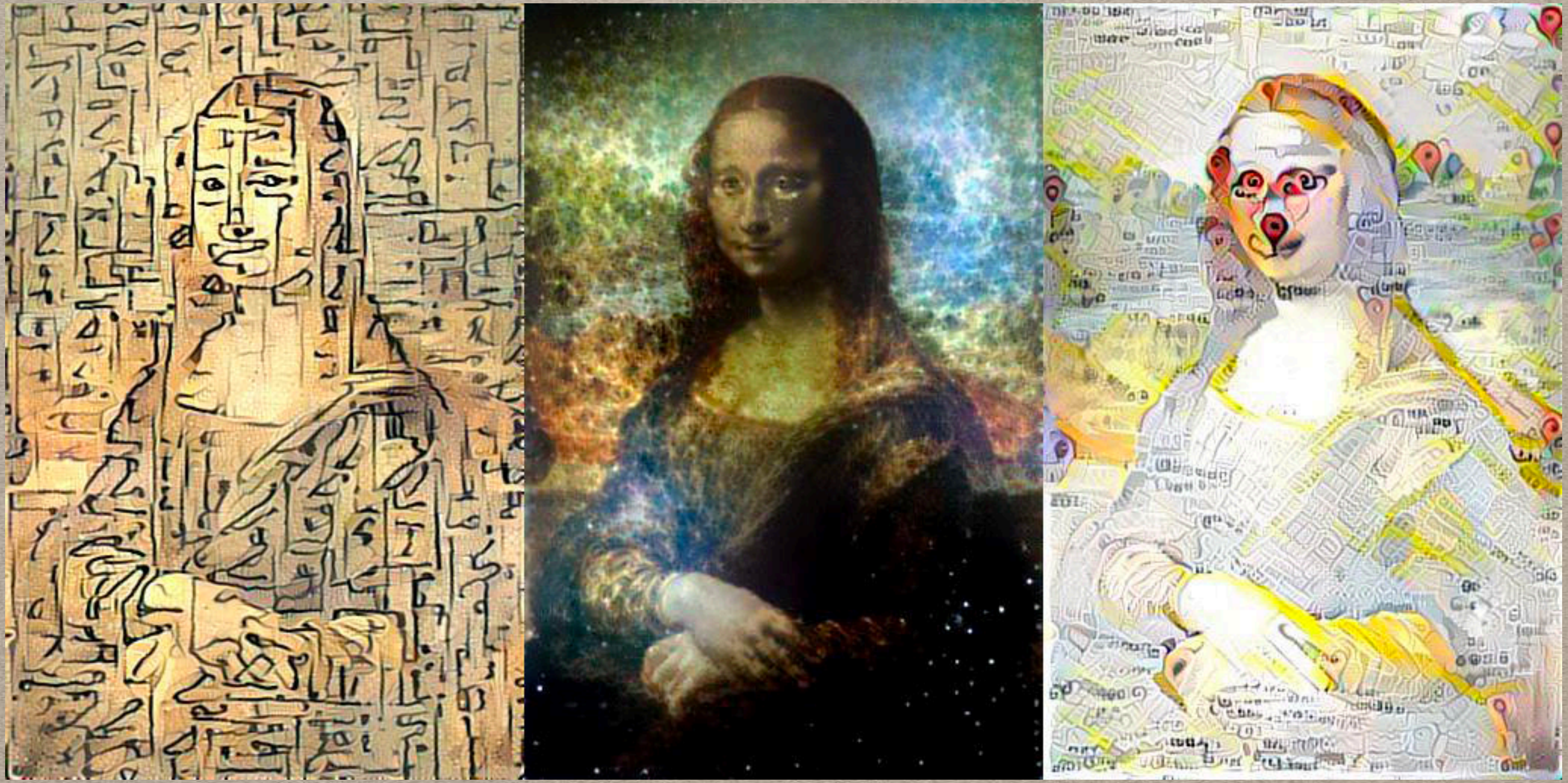



Horizon
Trees
Leaves
Towers & Pagodas
Buildings
Birds & Insects

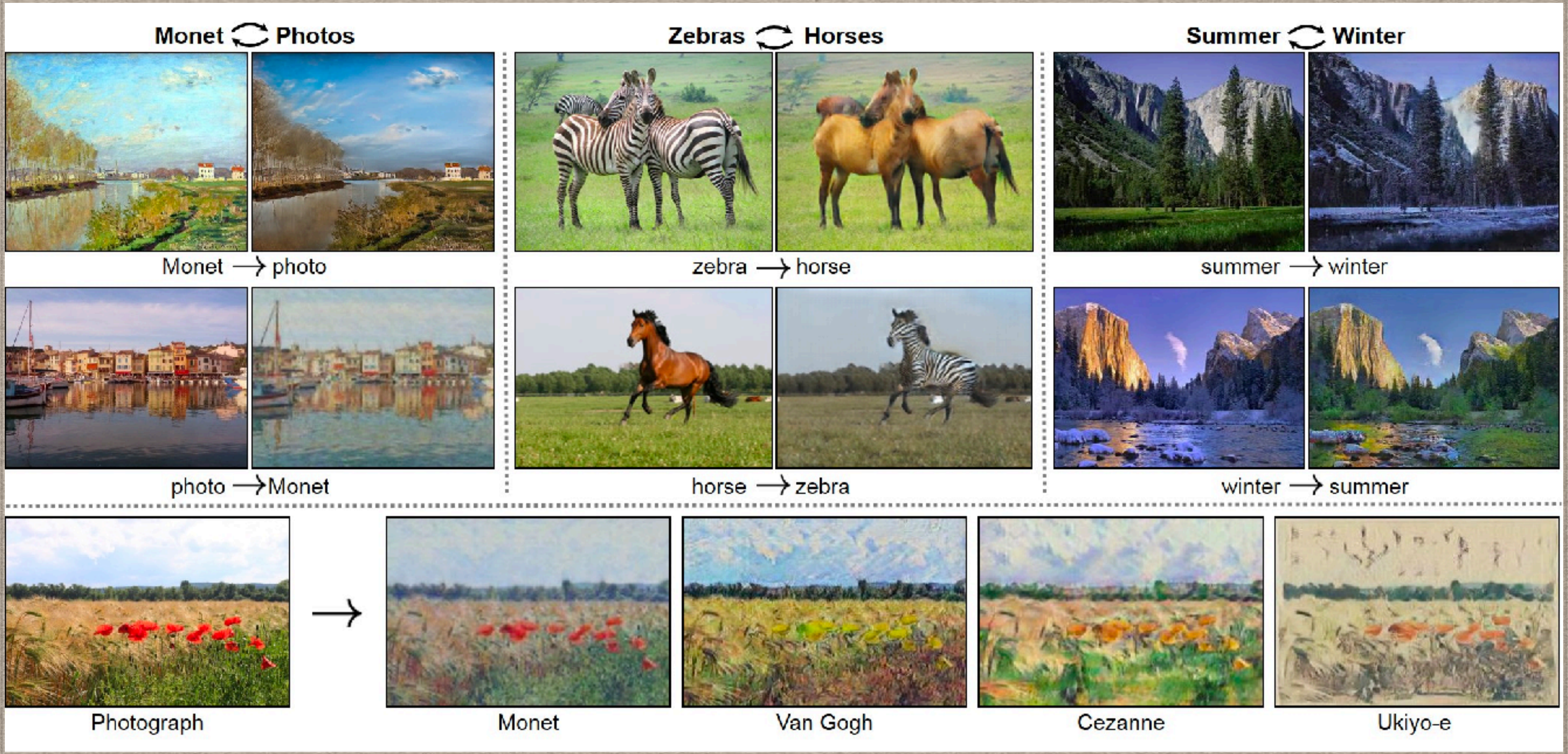
STYLE TRANSFER

https://raw.githubusercontent.com/fzliu/style-transfer/master/images/style/starry_night.jpg





http://genekogan.com/works/style-transfer/



https://junyanz.github.io/CycleGAN/images/teaser_high_res.jpg

NO FREE LUNCH

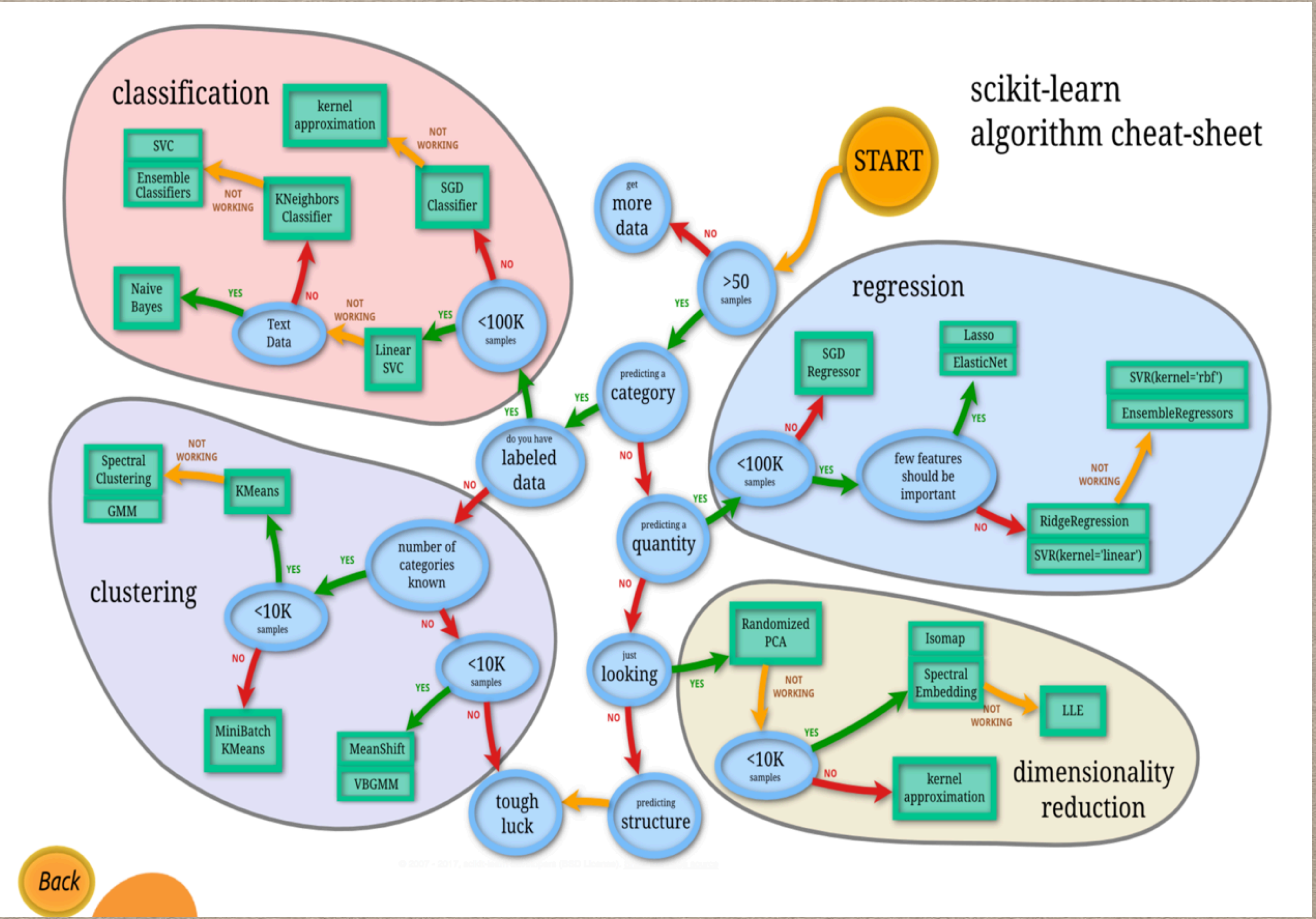
- Classifier performance regresses to the mean
- Any classifier that outperforms on some datasets must underperform on others
- No general purpose classifier
- Conservation of performance

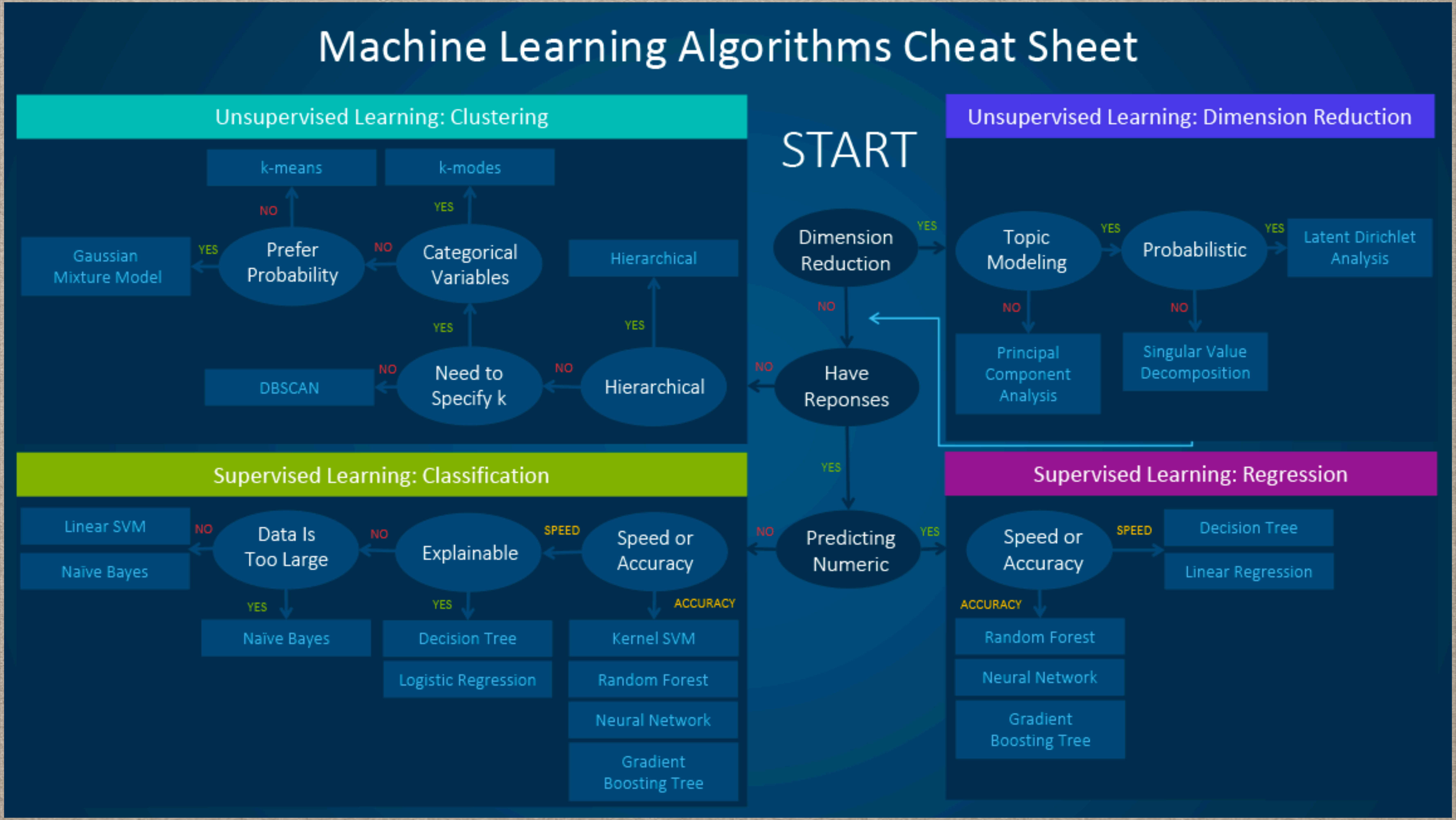
CURSE OF DIMENSIONALITY

- Spaces get sparse for each dimension added
- Amount of training data needed for statistical significance also grows

STEPS IN DEVELOPING A MODEL

1. Exploration
2. Algorithm selection
3. Feature engineering
4. Validation
5. Hyperparameter tuning


scikit-learn
algorithm cheat-sheet
START
>50 samples
get more data
predicting a category
do you have labeled data
<100K samples
SGD Classifier
kernel approximation
Linear SVC
Text Data
Naive Bayes
KNeighbors Classifier
SVC
Ensemble Classifiers
classification
number of categories known
<10K samples
KMeans
Spectral Clustering
GMM
MiniBatch KMeans
MeanShift
VBGMM
clustering
tough luck
predicting a quantity
regression
<100K samples
SGD Regressor
few features should be important
Lasso
ElasticNet
RidgeRegression
SVR(kernel='linear')
SVR(kernel='rbf')
EnsembleRegressors
just looking
predicting structure
Randomized PCA
<10K samples
Isomap
Spectral Embedding
LLE
kernel approximation
dimensionality reduction
YES
NO
NOT WORKING
Back


Machine Learning Algorithms Cheat Sheet
Unsupervised Learning: Clustering
k-means
k-modes
NO
YES
Gaussian Mixture Model
YES
Prefer Probability
NO
Categorical Variables
Hierarchical
YES
YES
DBSCAN
NO
Need to Specify k
NO
Hierarchical
START
Unsupervised Learning: Dimension Reduction
Dimension Reduction
YES
Topic Modeling
YES
Probabilistic
YES
Latent Dirichlet Analysis
NO
NO
NO
Principal Component Analysis
Singular Value Decomposition
NO
Have Reponses
YES
Supervised Learning: Classification
Linear SVM
Naïve Bayes
NO
Data Is Too Large
NO
Explainable
SPEED
Speed or Accuracy
YES
YES
ACCURACY
Naïve Bayes
Decision Tree
Logistic Regression
Kernel SVM
Random Forest
Neural Network
Gradient Boosting Tree
NO
Predicting Numeric
YES
Supervised Learning: Regression
Speed or Accuracy
SPEED
Decision Tree
Linear Regression
ACCURACY
Random Forest
Neural Network
Gradient Boosting Tree

COMMON ML ALGORITHMS

- NN - Neural Network
- DT - Decision Tree
- SVM - Support Vector Machine
- NB - Naive Bayes
- KNN - K-nearest neighbors
- Regression
- RF - Random Forest (*Ensemble* method)

FEATURE SELECTION

- Information gain
- scikit-feature
- Avoid the curse of dimensionality, long training times
- A form of dimensionality reduction

FEATURE ENGINEERING

- Pandas
- Apache Arrow

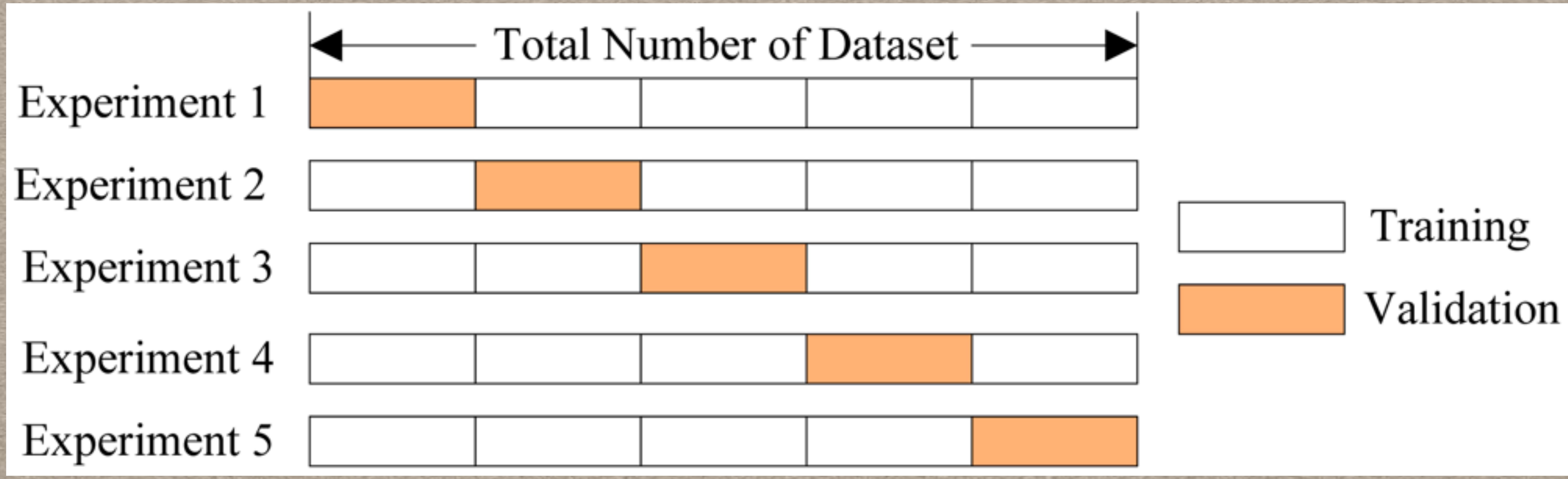
VALIDATION

- Training
- Test
- Validation

X-FOLD CROSS-VALIDATION

FIVE FOLDS

https://www.kaggle.com/dansbecker/cross-validation



HYPER-PARAMETER TUNING

- AutoML
  - Grid search - http://scikit-learn.org/stable/modules/grid_search.html#grid-search
  - Tpot - https://github.com/EpistasisLab/tpot
  - Auto-sklearn - https://github.com/automl/auto-sklearn

FACEBOOK PROPHET GOALS

- Turnkey predictive modeling of time series
- Seasonalities, holidays
- Missing data, outliers
- Historical trend changes
- Logistic growth and decay

FACEBOOK PROPHET IMPLEMENTATION

- Additive regression
  - Piecewise linear or logistic growth trend
  - Yearly seasonal component (Fourier series)
  - Weekly seasonal component (dummy variables)
  - Holidays (user-provided)

EXAMPLE II: TIME SERIES REGRESSION WITH PROPHET

ML FOR TRADING

- Simulated trading - Reinforcement learning can find an arbitrage edge in the data [RMLT17]
- Time series classification
  - KNN, Logistic regression at ~80% accuracy (better on longer timescales) [FTS17]
  - Time series regression - Linear Regressor, Support Vector Regressor at .016 RMSE [FTS17]
  - NN and Gaussian Process best [ECML10]
- News sentiment analysis - mixed results [SAN11]; some success for SVMs (~80%) [SPNS]

PRE-PROCESSING IS KEY [ECML10]

- De-seasonalization
- Log transformation
- First-order differences
- Moving averages
- Windowing

TECHNIQUES FOR TIME SERIES [FTML10]

Technology	Number	Publications
ANN based	21	[1], [4], [5], [8], [13], [15], [16], [20], [24], [25], [27], [31], [33], [35], [36], [37], [38], [39], [41], [43], [46]
Evolutionary & optimisation techniques	4	[23], [29], [30], [45]
Multiple / hybrid	15	[2], [3], [6], [7], [11], [14], [17], [18], [21], [22], [26], [32], [34], [40], [42]
Other	6	[9], [10], [12], [19], [28], [44]

Table 1: Reviewed papers classified by machine learning technique

DATA SIGNALS FOR TIME SERIES [FTML10]

Input	Number	Publications
Lagged Index Data	35	[1], [2], [3], [4], [5], [6], [7], [8], [9], [11], [13], [14], [15], [16], [17], [19], [21], [24], [25], [26], [27], [28], [31], [33], [34], [35], [36], [37], [38], [39], [41], [42], [44], [45], [46]
Trading Volume	4	[11], [25], [28], [46]
Technical Indicators	13	[3], [4], [10], [20], [22], [23], [28], [29], [30], [32], [40], [41], [43]
Oil Price	4	[12], [15], [33], [38]
S&P 500 / NASDAQ / Dow Jones (non US studies)	4	[18], [20], [33], [41]
Unemployment Rate	1	[38]
Money Supply	3	[12], [38], [39]
Exchange Rates	3	[15], [18], [41]
Gold Price	3	[12], [15], [33]
Short & Long Term Interest Rates	6	[5], [15], [25], [26], [35], [39]
Others	10	[4], [5], [15], [17], [20], [26], [35], [38], [39], [41]

Table 3: Reviewed papers classified by input variables

TIME FRAMES FOR TIME SERIES [FTML10]

Time-frame	Number	Publications
Day	31	[1], [2], [3], [4], [6], [7], [8], [9], [10], [13], [14], [17], [19], [20], [21], [22], [24], [27], [28], [31], [32], [33], [34], [35], [36], [37], [40], [41], [42], [44], [45]
Week	3	[18], [23], [43]
Month	3	[26], [38], [39]
Multiple / Other	9	[5], [11], [12], [15], [16], [25], [29], [30], [46]

Table 2: Reviewed papers classified by forecasting time-frame

EVALUATION OF TIME SERIES [FTML10]

Eval. Model	Number	Publications
Buy & Hold	9	[3], [4], [5], [18], [25], [38], [39], [41], [43]
Random Walk	6	[5], [11], [18], [22], [28], [39]
Statistical Techniques	18	[5], [6], [9], [10], [11], [13], [15], [17], [18], [19], [24], [26], [28], [34], [35], [37], [39], [41]
Other Machine Learning Techniques	28	[2], [3], [4], [6], [7], [8], [11], [13], [14], [17], [18], [21], [22], [23], [24], [26], [29], [30], [31], [32], [34], [35], [39], [40], [42], [44], [45], [46]
No Benchmark Model	7	[1], [12], [16], [20], [27], [33], [36]

Table 4: Reviewed papers classified by evaluation models

TLDR - ML FOR TRADING

- It’s early; no clear winner
- Consider NNs (directional, classification), SVMs (sentiment classification), Linear and SV regressors (regression)

KEY TECH

- Docker
- GitHub
- Jupyter
- Quilt
- Reproducibility - "It's so bad it sometimes feels like stepping back in time to when we coded without source control." -Pete Warden

RECOMMENDED LIBRARIES & BOOKS

- Use Keras to get started with DNNs
- Use GPUs (or TPUs) to reduce training time
- Use scikit-learn to explore different algorithms
- Use Prophet for time series forecast
- Recommended Reading: *Deep Learning: A Practitioners Approach*

REFERENCES AND CODE SAMPLES

HTTPS://GITHUB.COM/QUILTDATA/REPRODUCIBLE-ML/