

## Practical Examples of Encoding Structure

1. [Attentional ShapeContextNet for Point Cloud Recognition](#)
2. [Aggregated Residual Transformations for Deep Neural Networks](#)
3. [Why does deep and cheap learning work so well?](#)
4. [Symmetry Regularization](#)
5. [Group Equivariant Convolutional Networks](#)
6. [Spatial Transformer Networks](#)
7. [Stochastic Video Generation with a Learned Prior](#)
8. [Relational inductive biases, deep learning, and graph networks](#)
9. [Distant Transfer for Continual Learning \[Marc Pickett\]](#)
10. [Relational Deep Reinforcement Learning](#)

I'm extremely surprised that I haven't seen this comprehensive mode of thinking made explicit anywhere. If I'm actually the first person to come up with this abstraction (and the surprise of & questions asked by people like Pedro Domingos, Ryan Adams suggests that it's rare) then I have a serious duty at hand. This is key to algorithm learning and creating low bias models against the curse of dimensionality. This in an ensemble model.

Types of structure in information:

1. Hierarchical / Compositional / Combinatorial Structure
2. Relational / Graphical Structure
3. Recursive Structure
4. Temporal / Sequential Structure
5. Clustering Structure
6. Discreteness - quantized
7. Continuity - distribution
8. Smoothness
9. Sparsity
10. Locality
11. Linearity / Polynomial / Exponential Structure

Principles of Structure:

1. Simplicity vs. complexity
2. Bias - Variance Decomposition
3. Abstraction - level of abstraction at which more or less structure, or different types of structure are present
4. Framed as Compression
  - a. Degree of Compression
5. Directionality
6. Discrete vs. Continuous
7. Abstraction - fine vs. coarse grain structure

8. Similarity, say, with a feature or set of features
9. Randomness, degree to which there is structure, compressibility of data
10. Homogeneity - degree to which the same operations can be run over objects in the structure
11. Dimensionality - Interactions between features vs. single feature structure

### Examples

1. Hierarchical / Compositional / Combinatorial
  - a. Images
  - b. Language
  - c. Set of axioms to euclidean geometry
  - d. Organization's' management structure
2. Relational / Graphical
  - a. Social Network
  - b. Worldview (Tension with Hierarchical)
3. Recursive (Top-down hierarchical)
  - a. Trees
4. Temporal
  - a. Periodicity
  - b. Messages' bursting structure
  - c. Quantized, like hitting lights for predicting arrival time
  - d. Making food in a kitchen (Tempo)
  - e. Dancing (Rhythmic / Periodic)
  - f. Option / Permanence - School choice, Tattoos, Relationships
5. Discreteness
  - a. Categories - Number of Fields in an Academy
  - b. Binary - Graduated or Not Graduated, Accepted or not Accepted, Given an offer or Not Given an Offer
6. Continuity
  - a. Intensity of emotion
  - b. Amount of time on a task
7. Causal
  - a. Counterfactual - If I had done x, simulation.
  - b. Imagination - If I do x, simulation.

### Hierarchical Structure

1. Abstraction
2. Images
  - a. Objects - Object Parts - Shapes - Lines / Curves
3. Audio

- a. Words - Phonemes
- 4. Businesses / Governments
- 5. Sciences
  - a. Physics
  - b. Chemistry
  - c. Biology
    - i. Ontology of Species
    - ii. Organ Systems - Organs - Tissues - Cells - Nuclei + Organelles
    - iii. Brain
- 6. Natural Language
  - a. Fields - Concepts - Words (Combinatorial as well)
  - b. Paragraph - Sentence - Phrase - Word - Character
- 7. Time
  - a. Centuries - Decades - Years - Months - Weeks - Days - Hours - Minutes - Seconds
- 8. Measurement
  - a. Kilometers - Meters - Centimeters - Millimeters
- 9. Object Oriented Systems
  - a. Classes - Objects
- 10. Economy
  - a. GDP - consumer spending + investment + Government Spending + Exports - Imports

#### Relational / Graphical Structure

- 1. Object Oriented Structure
  - a. Object (Entity)
  - b. X is a Y relationships (Classification, Inheritance)
  - c. X has a Y relationships (Composition / Aggregation)
  - d. Properties of an Object
- 2. Causal Graph - X leads to Y
- 3. Dependency - X depends on Y
- 4. Subject - Object relationships (in sentences)
  - a. Linking verbs - 'is', 'has', 'are', 'being', 'sense' etc. between Object and Subject
- 5. Co-occurrence
  - a. Ex. Words mentioned in concert with one another
- 6. Link - are connected
  - a. Linkage Distribution
- 7. Locality
- 8. Edge Density

#### Temporal Structure

1. Periodicity
  - a. Hierarchical Periodicity
  - b. Seasonality
2. Burstiness
3. Stationary vs. Non-Stationary Distributions
4. Permanence / Option Structure
5. Quantized
  - a. Ex. hitting lights when predicting arrival time
6. Autoregression / Autocovariance
7. Feedback
  - a. Positive Feedback
  - b. Negative Feedback
  - c. Length of feedback loops
8. Synchronicity vs. Asynchronicity
  - a. Discrete vs. Continuous
9. Exponential Decay vs. Windowing
  - a. Continuity vs. Discreteness
10. Stability & Equilibrium
11. Derivatives - change over time
12. Objectness - these pixels move together
13. Asymmetry between past and future
14. Exclusive ability to directly impact present
15. Strong predictor of causality / anti-causality

#### Relevant Links

1. <https://sites.google.com/site/icml18limitedlabels/>
2. <https://arxiv.org/pdf/1608.08225.pdf>

#### Papers

- 1.

#### Notes

Regularizers impose a smoothness inductive bias, and weight decay / L2 regularization happens to impose smoothness.

But at the end of the day, we do induction. We realized that this bias worked well in the past and impose it on new data.

Big diff between having a causal model (true relationship is smooth, so imposing that prior will lead to a more efficient search in function space) and just predicting that it will work well because it worked on past data (with no model for why it's working)

There are different inductive biases imposed by every form of regularization, which should be listed and maximized.

1. Dropout
  - a. Algorithmic. Cuts the signal for inputs to a network.
    - i. Does dropout work for linear models? For trees? How to deal with it at test time?
2. Norm Penalties
  - a. L1 (Sharp)
  - b. L2 (Smooth) / Weight Decay
3. Model averaging
  - a. (the averaging step, not the step where variance is created through bagging, alternate parameterization, feature elimination (al rf & erf), etc.
4. Intelligent Initialization
5. Noise Injection
6. Early Stopping
7. Constraints on optimization
8. Train and test time data augmentation
9. Multi-task learning
  - a. Multi-class as multi-task
10. Pruning
11. Weight Sharing
12. Stochastic Optimization
13. All models? All Priors?