# Introduction to Machine Learning

Wray Buntine
Monash University
http://Bayesian-Models.org

6th February, 2020

# ACML in Bangkok

- 18-20th November at Berkeley Hotel Pratunam
  - colocated with ICONIP
- cheap flight to Bangkok not much more than a flight to Brisbane
- always a high quality tutorial and invited speaker programme!

**Deadlines:**

- journal track (for ML Jnl) papers due 15th April
- conference track papers due 15th June
- workshop and tutorial proposals due mid July

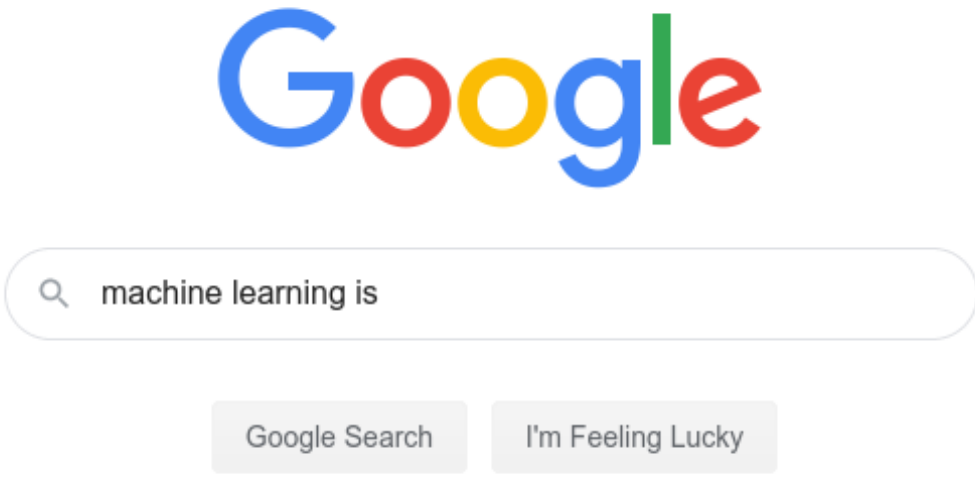
# Australian ML Research School

- 29th June to 1st July
- hosted at RMIT
  - near Central Station in Melbourne CBD
- intended for research students
- cheap registration for students!
- website and details TBD

# Background

- ML is now happening at a world-wind pace, with a phase shift happening a few years ago:
  - huge teams making rapid using extensive compute resources
  - results/methods already outdated and superceded at their point of publication
  - an "academic singularity" in a sense
- ML is widely recognised as a key technology component across many industries, also driving Data Science
- This is a big-picture/big-ideas talk.
  - no time for detail

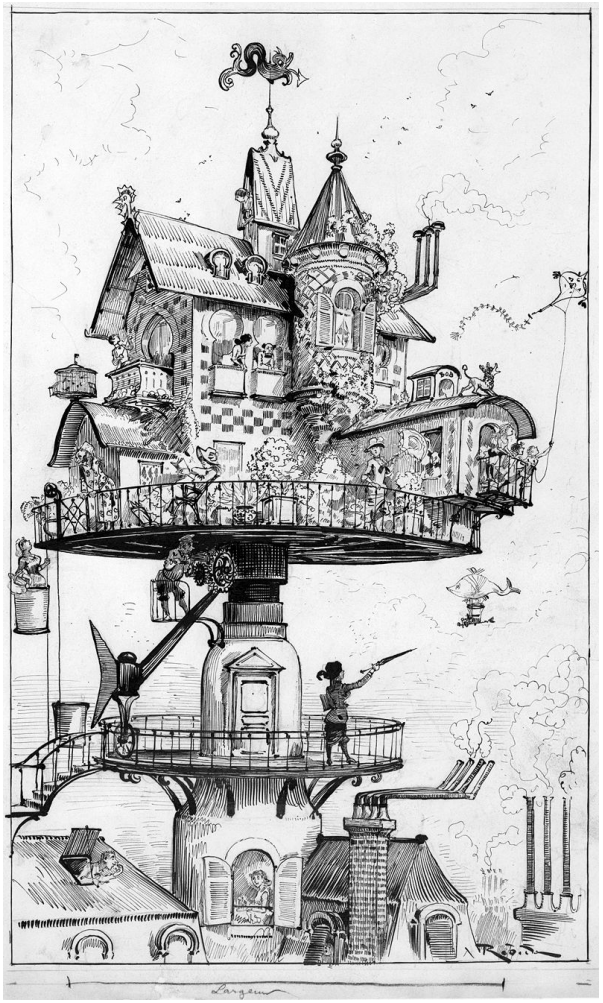
# Machine Learning is ...




*Machine Learning is ...*

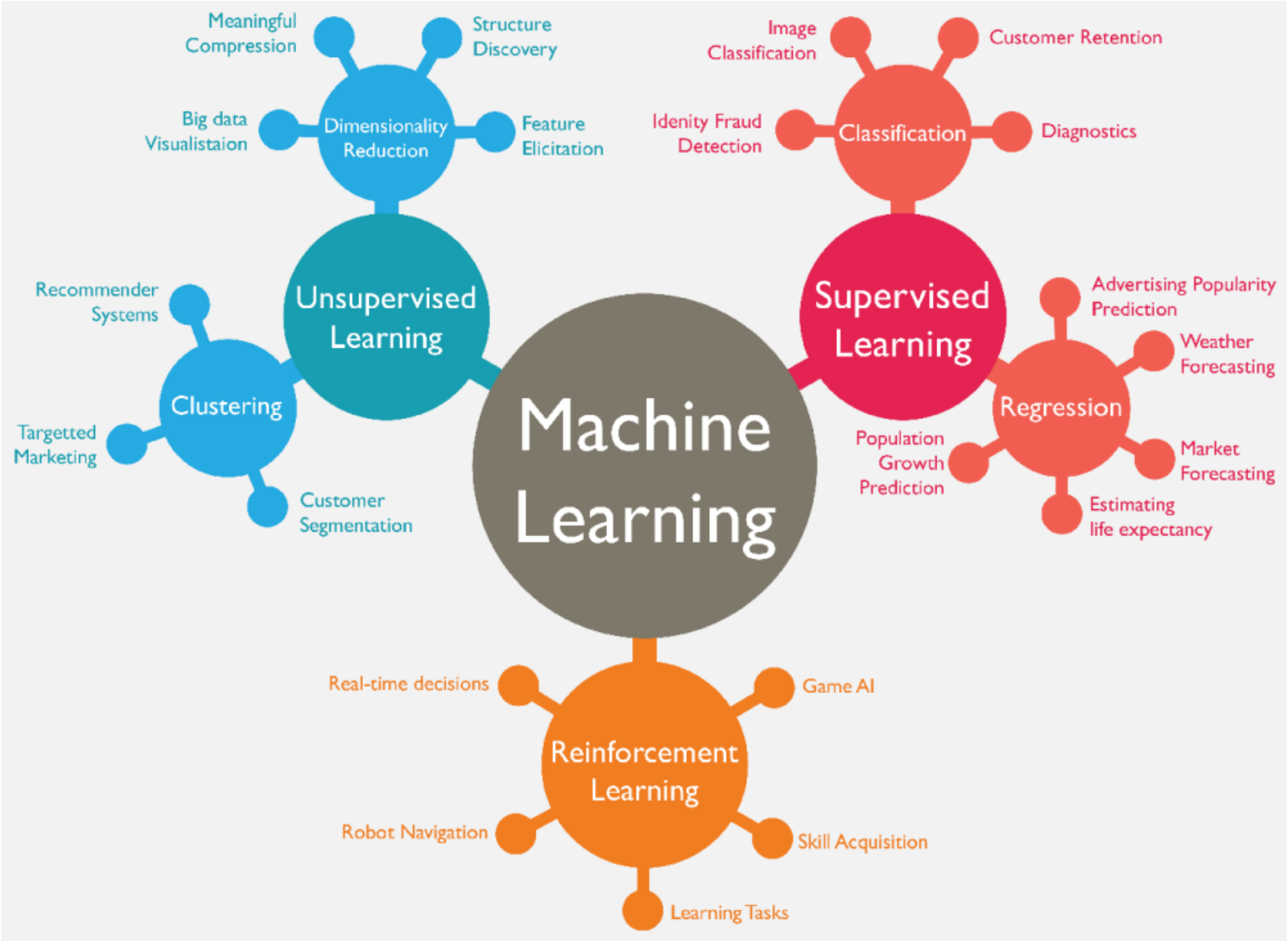
*Google Search Trends*

# Outline

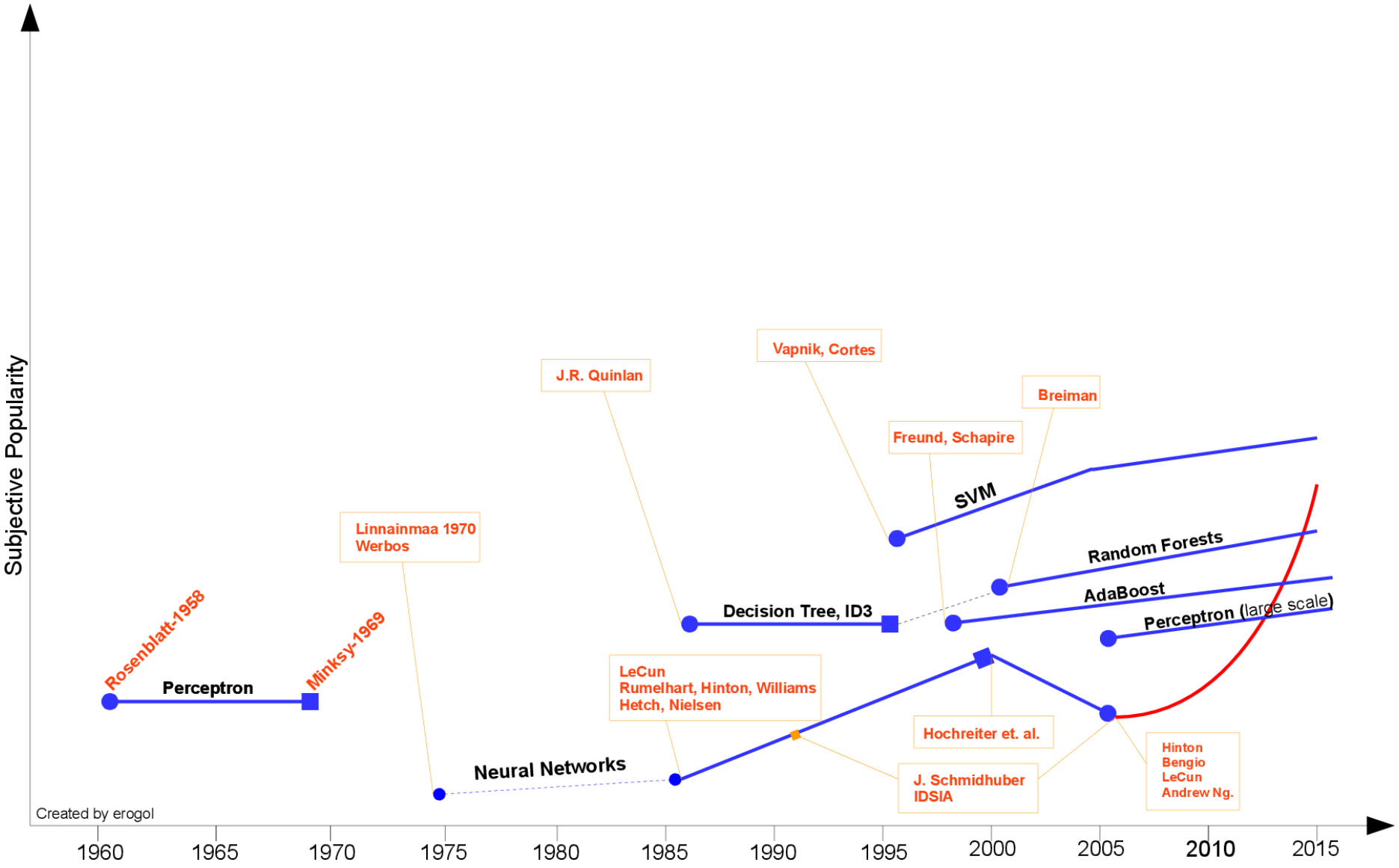


# What is Machine Learning?

(from Mactores Data Science Team)

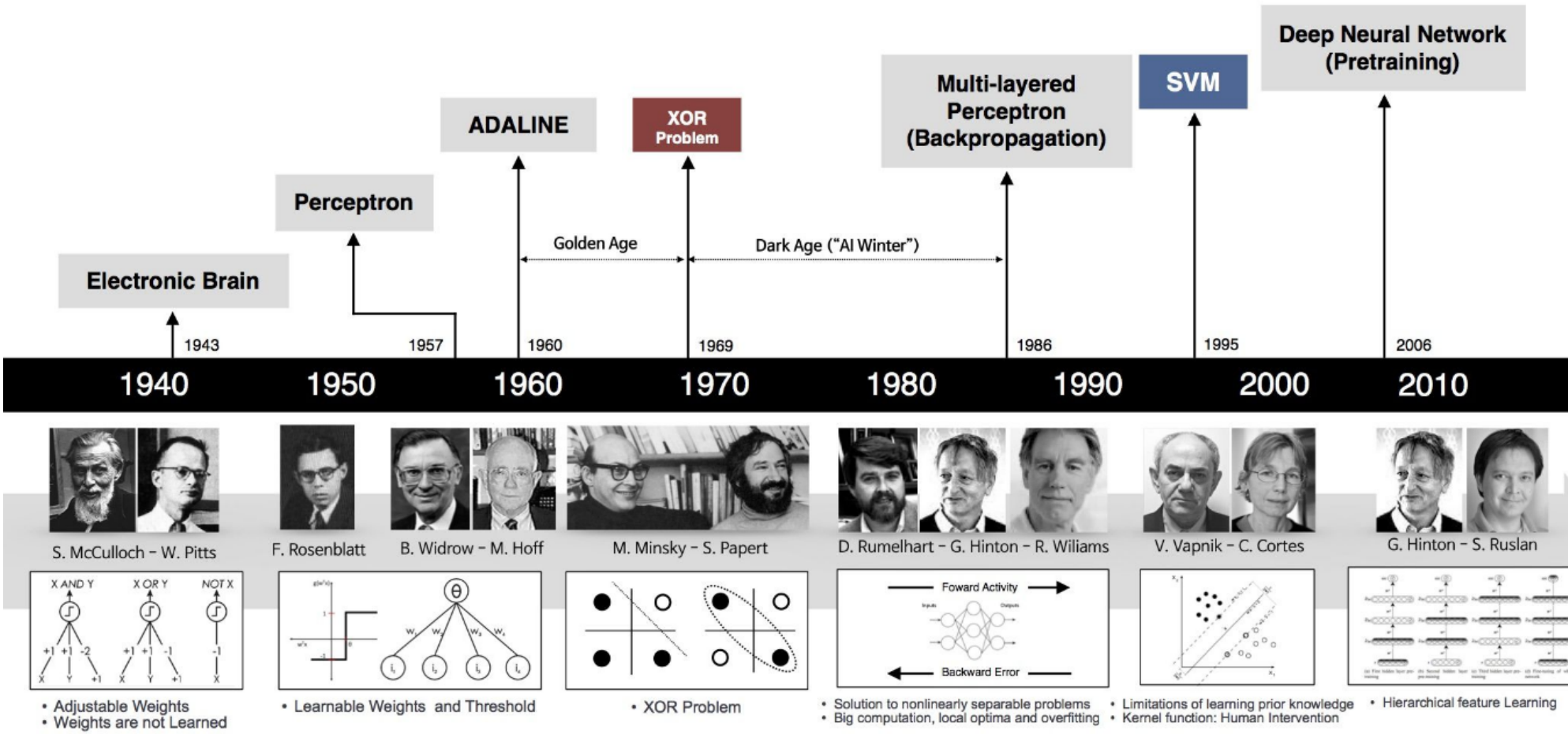


# Machine Learning Timeline



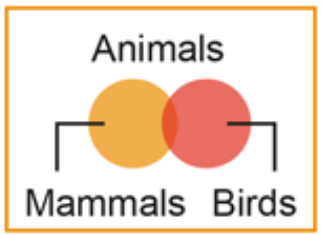
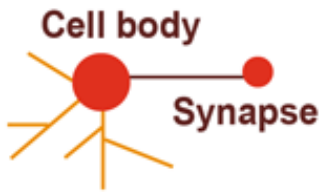

from `http://www.erogol.com/brief-history-machine-learning/`

# Deep Learning Timeline



# Machine Learning Tribes (circa 1990s)

***What are the five tribes?***

| Symbolists | Bayesians | Connectionists | Evolutionaries | Analogizers |
|---|---|---|---|---|
|   |   |   |  |  |
| Use symbols, rules, and logic to represent knowledge and draw logical inference | Assess the likelihood of occurrence for probabilistic inference | Recognize and generalize patterns dynamically with matrices of probabilistic, weighted neurons | Generate variations and then assess the fitness of each for a given purpose | Optimize a function in light of constraints ("going as high as you can while staying on the road") |
| **Favored algorithm** Rules and decision trees | **Favored algorithm** Naive Bayes or Markov | **Favored algorithm** Neural networks | **Favored algorithm** Genetic programs | **Favored algorithm** Support vectors |

Source: Pedro Domingos, *The Master Algorithm*, 2015

# ML Timeline: Conferences

AAAI: American Association of Artificial Intelligence, started **1980**

- the second attempt, an earlier one done in 1960s

ICML: International Conference on Machine Learning, started **1985**

NeurIPS: Neural Information Processing Systems started **1988**

—: in the interim ICML and NeurIPS merged in content and the dominant paradigm become *Bayesian neural networks and nonparametric methods*

ICLR: International Conference on Learning Representations, started **2012**

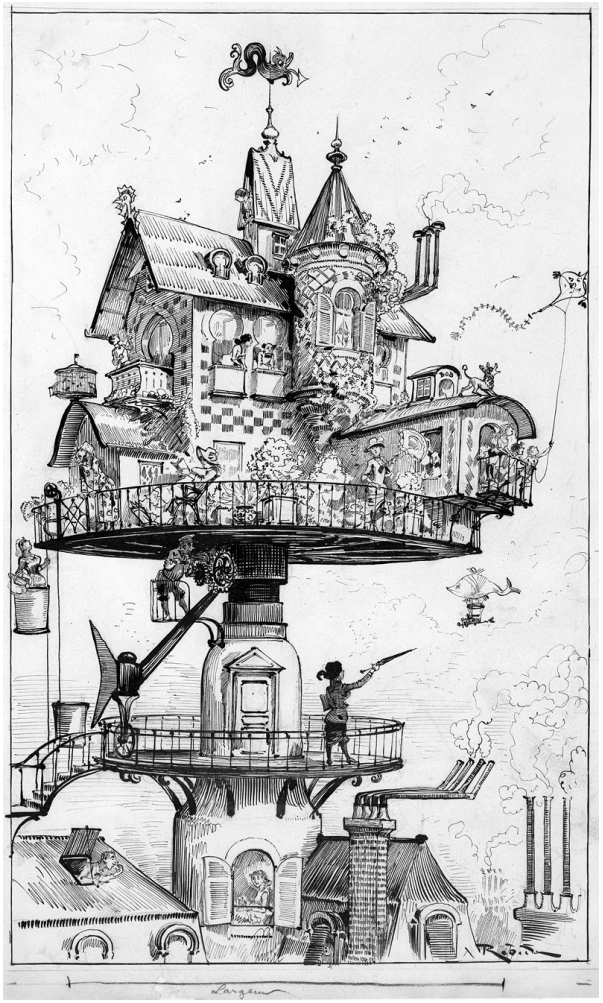
# ML Literature

- a lot of the good work comes from the application community: vision, media, natural language
- preference to publish in conferences
- most work appears on *arXiv.org* before publication
- so use appropriately linked search engines (e.g., *SemanticScholar.org* not the publisher ones (e.g., Web of Science, though this is great otherwise) to get arXiv cover

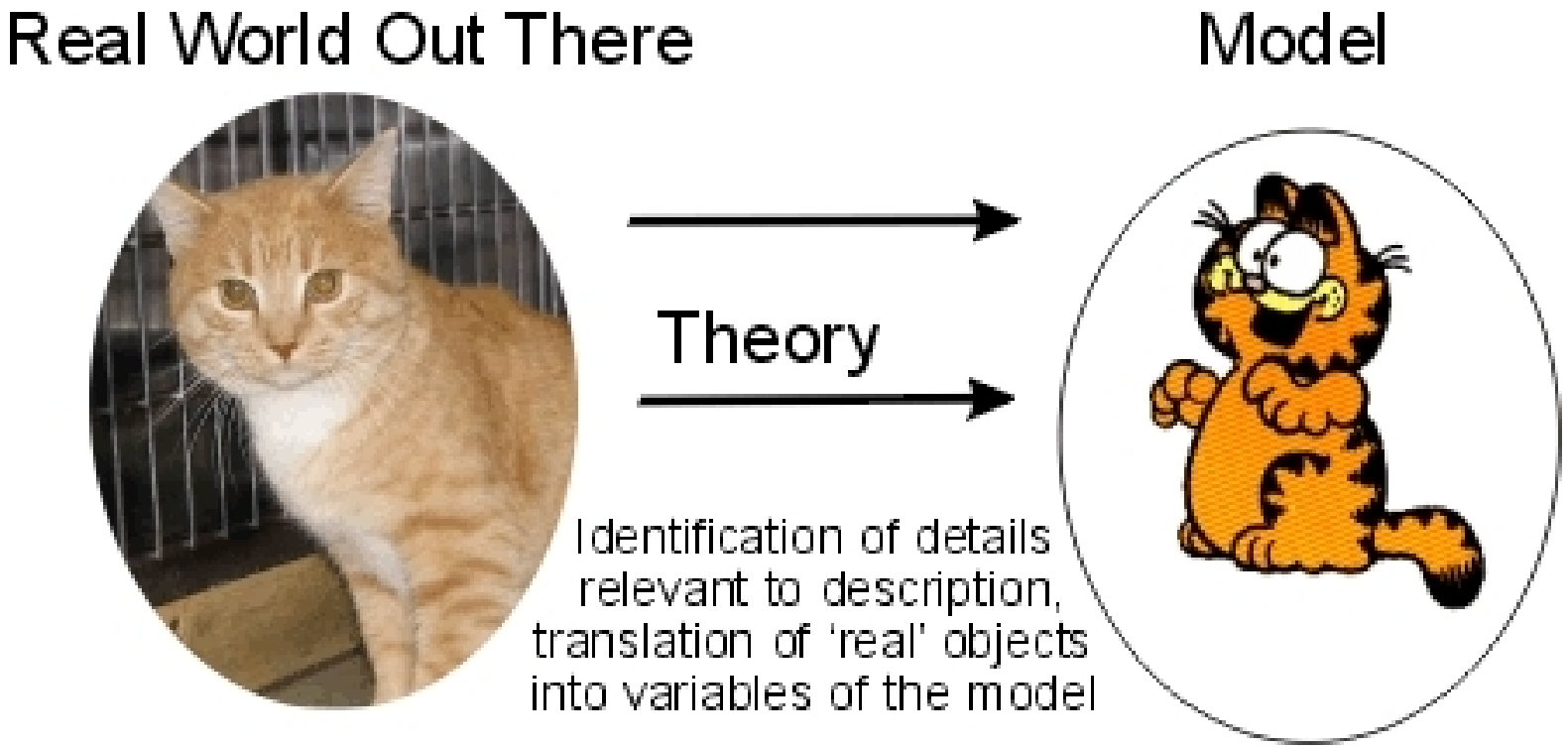
# Learning ML

- Python is the most common coding language
- generally need basic Probability, Linear Algebra and Calculus to understand
- moves so fast, so better to look at recent text books,
  - e.g. *Dive into Deep Learning*
    - most good books will be free online with PDF
- lots of great tutorials online
  - e.g. *Machine Learning Summer Schools*, browse recent ones and go looking for slides and videos
    - also at all the top conferences, ICML, NeurIPS, SIGKDD, ACL, CVPR, and many more, usually with videos online
- for simple but uptodate introductions targetting amateurs, see relevant sections on *Medium.com*
- Andrew Ng's courses on *Coursera.org* generally considered best MOOC intros

# Outline



# Models




from *the BackReaction blog by Sabine Hossenfelder*

but we focus on models that are predictive about out targets of interest

# Simple Example: Binomial Model

Task: estimate the frequency for a Boolean event given a sequence $S$ of length $N$ in the form $S \in \{T, F\}^N$

Model: probability of a single $T$ is $\theta$

Statistics: $n_T$ is number of $T$s in $S$, $n_F$ is number of $F$s in $S$

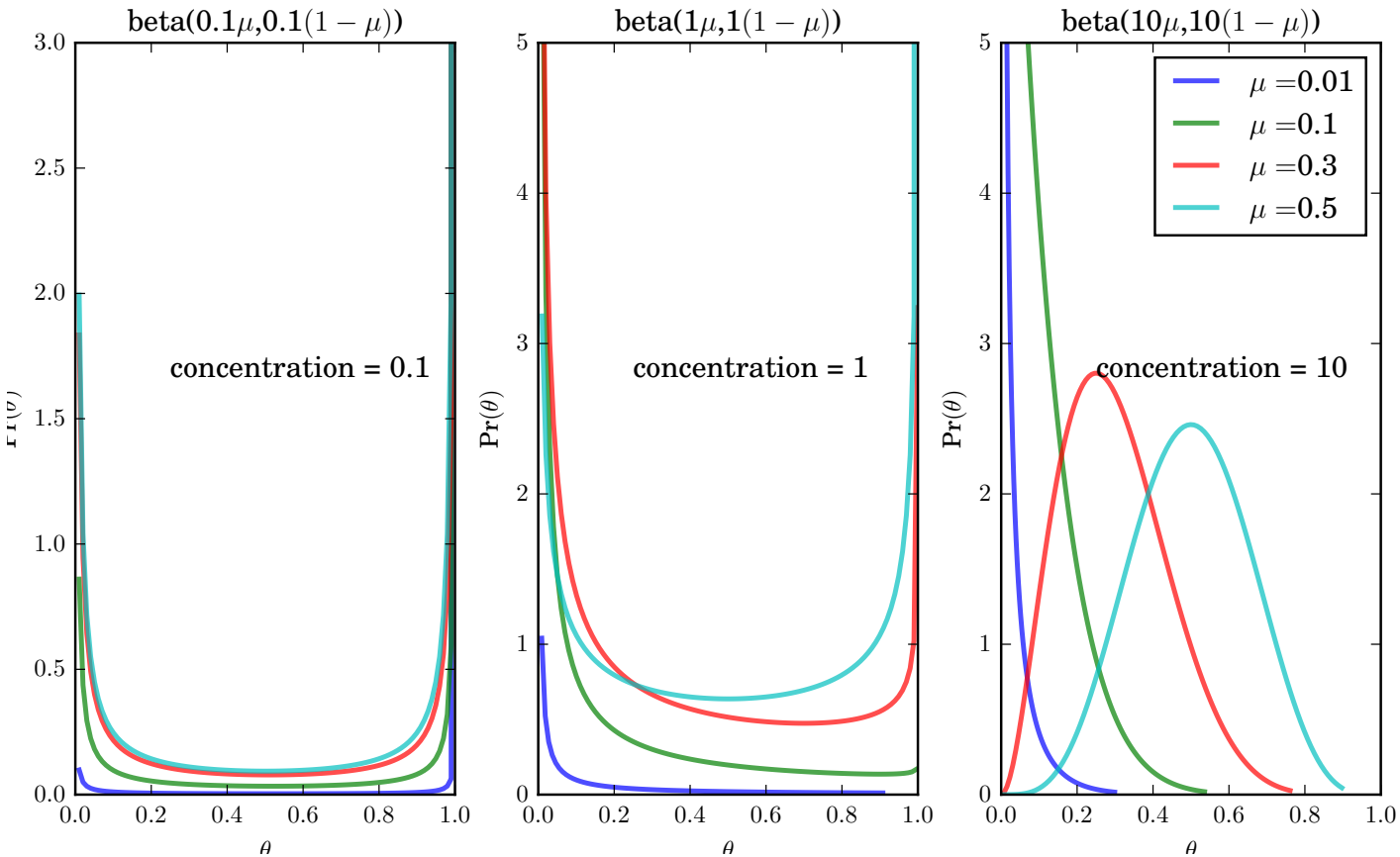
**Examples:**

- probability $\theta$ that you have a malignant melanoma
- probability $\theta$ that the top card from a deck is an ace

# Binomial Model Prior

$$p(\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}$$

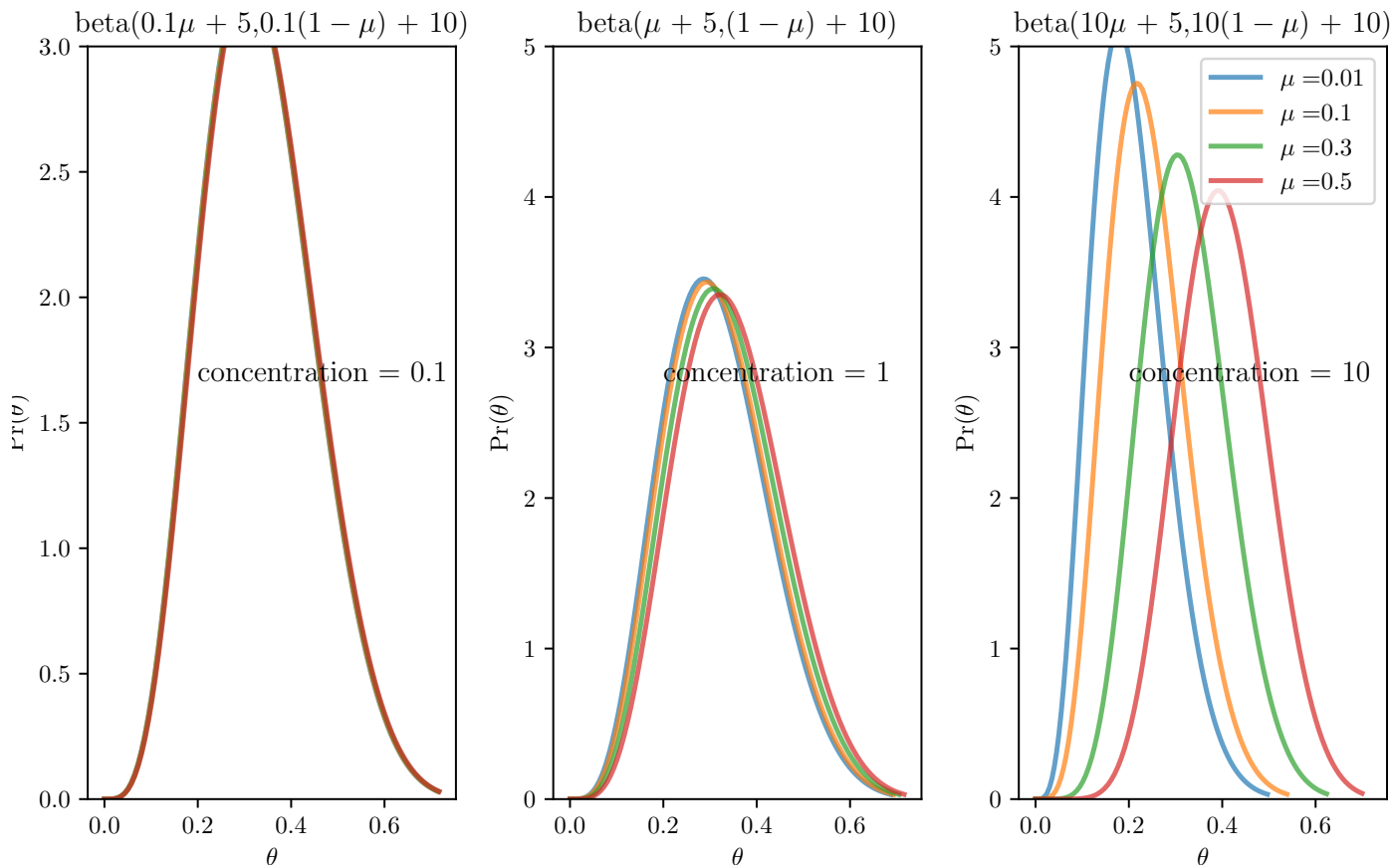
concentration="prior count" $= \alpha + \beta$, and mean$=\mu = \frac{\alpha}{\alpha+\beta}$.



# Binomial Model Inference

$$p(S,\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}\theta^{n_T}(1-\theta)^{n_F}$$

$$p(\theta|S) = \frac{\Gamma(\alpha+\beta+n_T+n_F)}{\Gamma(\alpha+n_T)\Gamma(\beta+n_F)}\theta^{\alpha+n_T-1}(1-\theta)^{\beta+n_F-1}$$



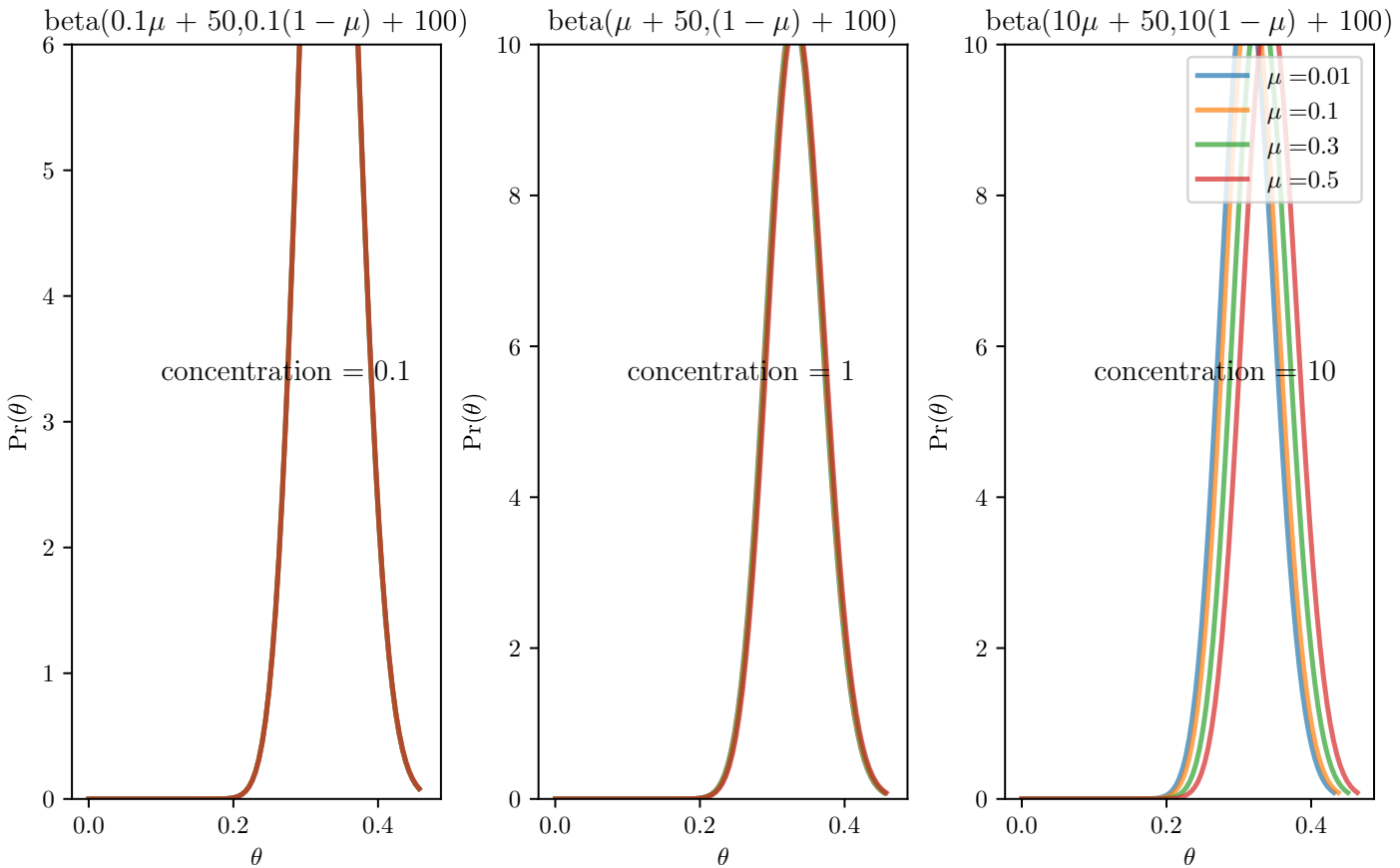

$n_T = 5$

$n_F = 10$

# Binomial Model Inference, cont.

$$p(S,\theta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}\theta^{\alpha-1}(1-\theta)^{\beta-1}\theta^{n_T}(1-\theta)^{n_F}$$

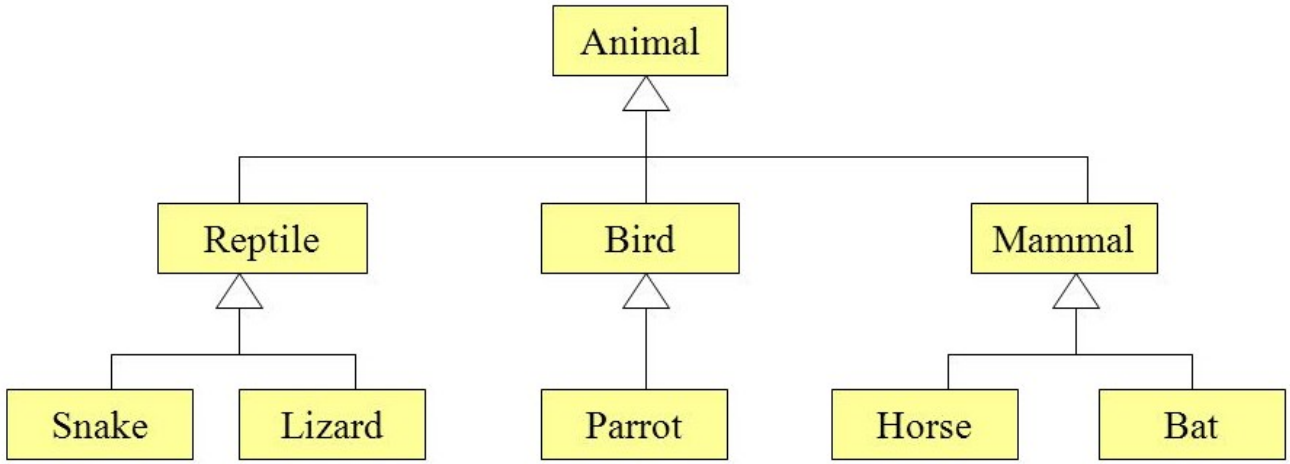
$$p(\theta|S) = \frac{\Gamma(\alpha+\beta+n_T+n_F)}{\Gamma(\alpha+n_T)\Gamma(\beta+n_F)}\theta^{\alpha+n_T-1}(1-\theta)^{\beta+n_F-1}$$




$n_T = 50$

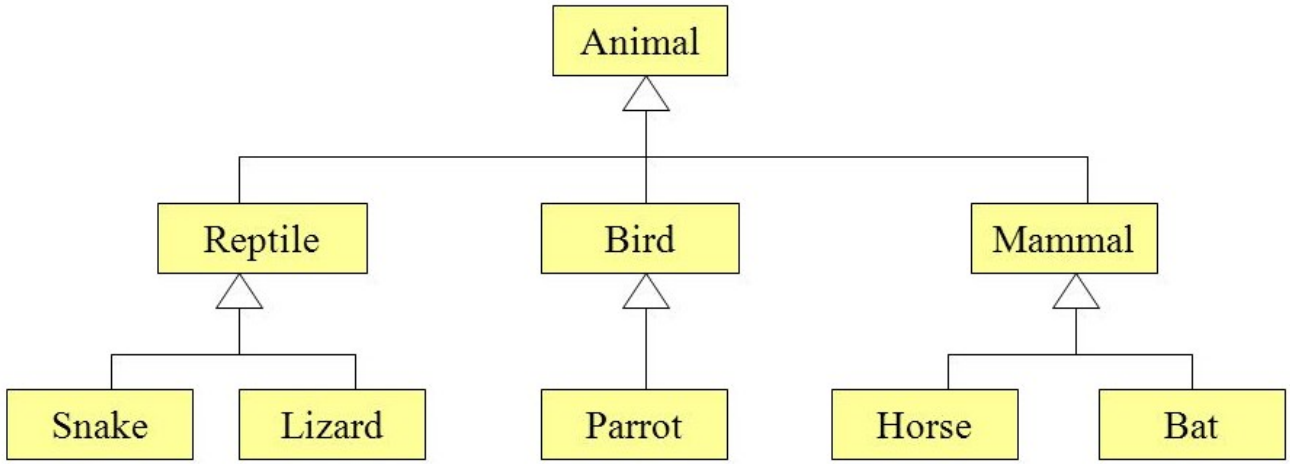
$n_F = 100$

# Hierarchical Models




- organise concepts into a hierarchy
- "share" model parts when learning to classify
  - e.g. recognising "leaves" is useful shared feature when learning to classify trees, shrubs, flowering plants, etc.
- how can we design models for different tasks to do "sharing"?

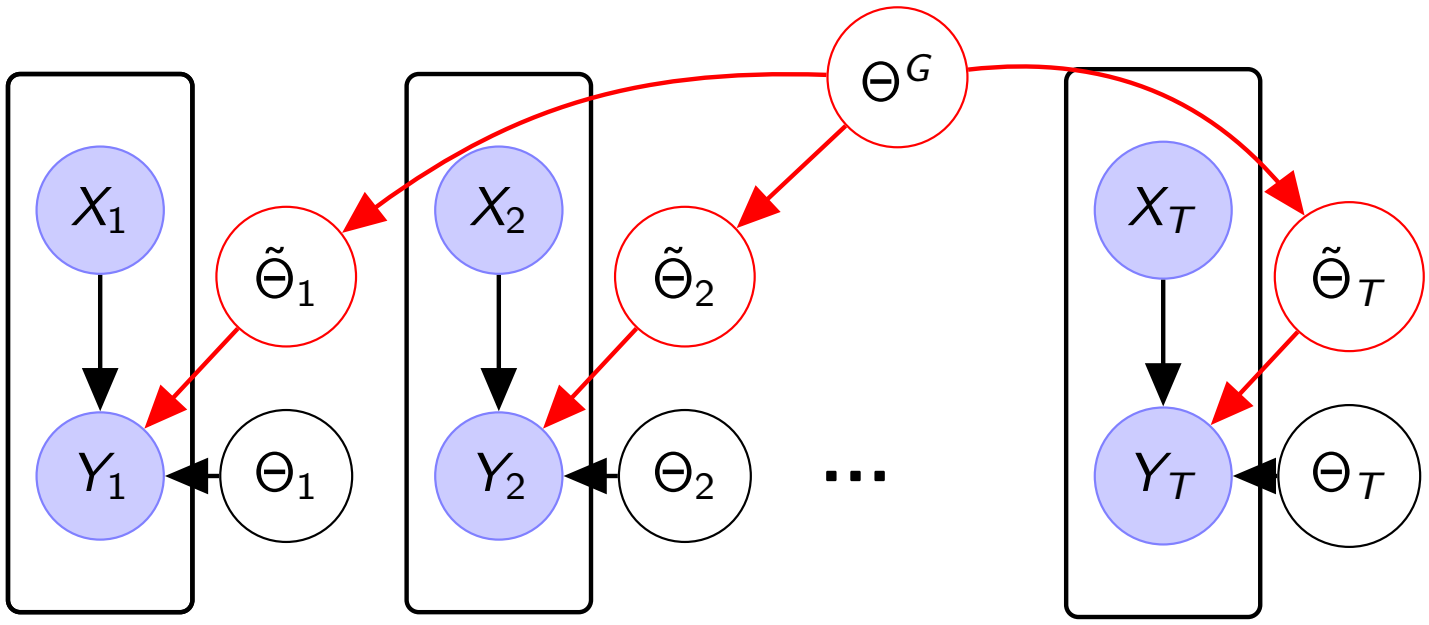
# Hierarchical Models




- organise concepts into a hierarchy
- "share" model parts when learning to classify
  - e.g. recognising "leaves" is useful shared feature when learning to classify trees, shrubs, flowering plants, etc.
- how can we design models for different tasks to do "sharing"?
- hierarchical Bayesian models implement the idea using a tree-structured Bayesian network of parameters corresponding to the hierarchy

# Hierarchical Bayesian Models

We have $T$ "related" problems with data $(X_t, Y_t)$.




$\Theta_1 - \Theta_T$: task specific parameters

$\Theta^G$: shared parameters

$\tilde{\Theta}_1 - \tilde{\Theta}_T$: task specific instantiation of shared parameters

# Knowledge-Based Reasoning


people use common-sense reasoning when doing classification:

- school bus (in America) is yellow and carries children
- a single adult with children may be their teacher

# Knowledge-Based Reasoning


people use common-sense reasoning when doing classification:

- school bus (in America) is yellow and carries children
- a single adult with children may be their teacher

- the kind of knowledge used is often different to that found in standard knowledge-bases like DBPedia

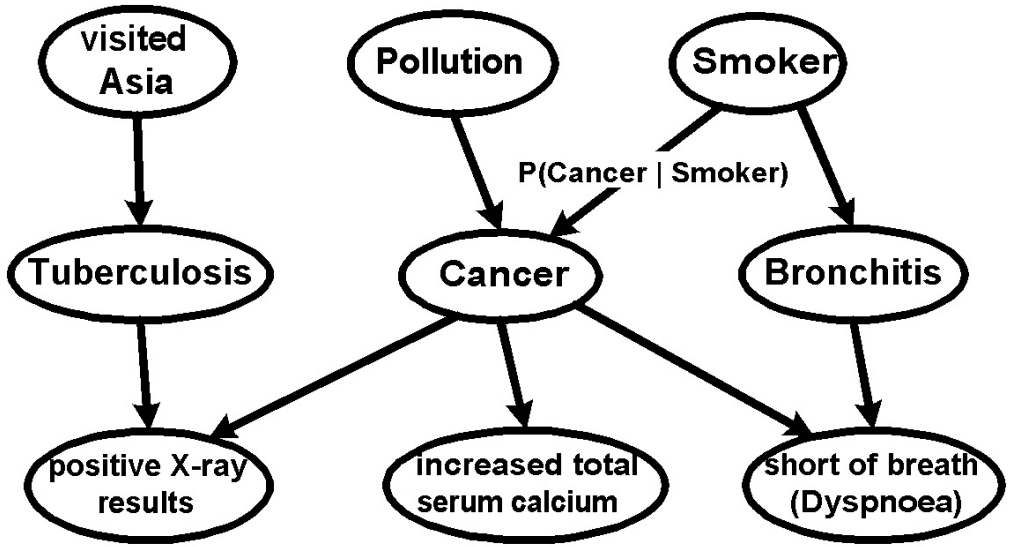
# Knowledge-Based Reasoning


people use common-sense reasoning when doing classification:

- school bus (in America) is yellow and carries children
- a single adult with children may be their teacher

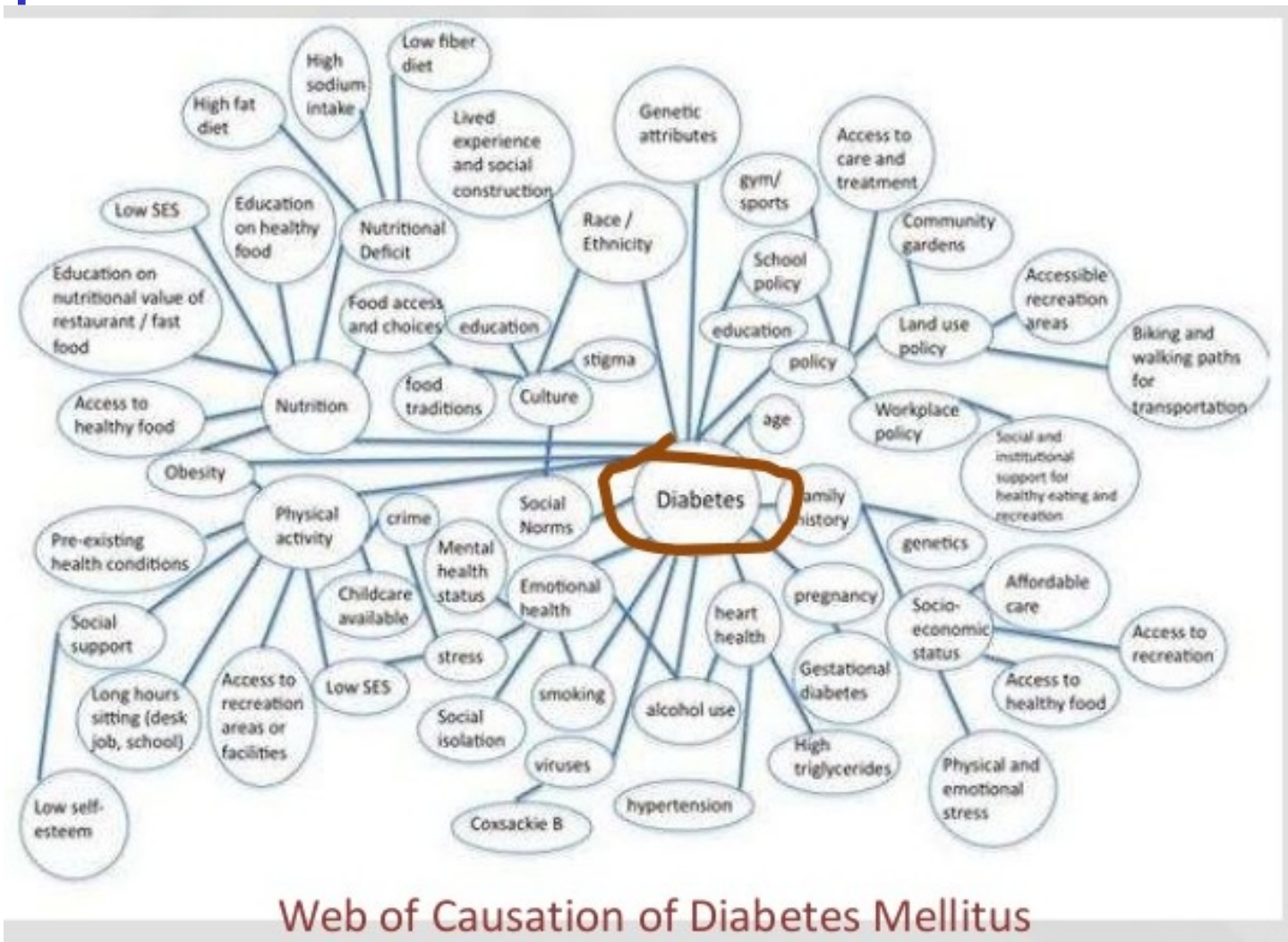
- the kind of knowledge used is often different to that found in standard knowledge-bases like DBPedia
- knowledge-bases are implemented as graphs in AI with objects as nodes and relationships as arcs

# A Simple Causal Model




**NB.** this famous figure is due to Lauritzen and Spiegelhalter, 1989
(don't blame me for implying Asia gives you tuberculosis!!)

# A Simple Correlational Model




from `https://www.slideshare.net/VishnuYenganti/association-causation`

**NB.** which relations are due to cause?

# Vision Understanding


How can we segment the image?

# Vision Understanding: Stereo


- Stereo gives us depth.
- Causality: neighbour pixels at different depth should be different objects, so segment.

  $\longrightarrow$ the woman is at a different depth to others, so segment off

# Vision Understanding: Stereo


- Stereo gives us depth.
- Causality: neighbour pixels at different depth should be different objects, so segment.

  $\longrightarrow$ the woman is at a different depth to others, so segment off

similar tricks can be applied to video with motion

# Using Causality: Example

**Task:** we want to collect data about car accidents. Which variables should we collect?

- external car color?
- color of seats?
- air-conditioning?
- quality of tyres?

# Using Causality: Example

**Task:** we want to collect data about car accidents. Which variables should we collect?

- external car color?
- color of seats?
- air-conditioning?
- quality of tyres?

When choosing variables, we try and imagine a causal connection from variable to cause of accident.

# Causality

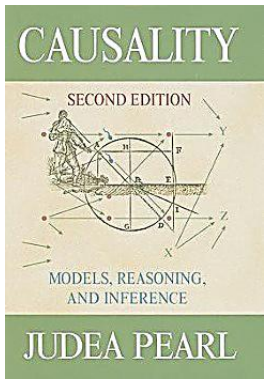
- causality allows much simpler models of the world
  - important technique for configuring learning problems in medicine and science
- humans are very good with causal reasoning
  - we infer causality based on time effects, controlled play, and the simplicity it provides in understanding the world

# Causality

- causality allows much simpler models of the world
  - important technique for configuring learning problems in medicine and science
- humans are very good with causal reasoning
  - we infer causality based on time effects, controlled play, and the simplicity it provides in understanding the world
- modelling causality is strongly related to modelling probabilistic independence
  - causal models yield the most comprehensive independence statements
- involves the capability of modelling intervention, externally fixing variables
  - rather than just passive observation

# Causality

- causality allows much simpler models of the world
  - important technique for configuring learning problems in medicine and science
- humans are very good with causal reasoning
  - we infer causality based on time effects, controlled play, and the simplicity it provides in understanding the world
- modelling causality is strongly related to modelling probabilistic independence
  - causal models yield the most comprehensive independence statements
- involves the capability of modelling intervention, externally fixing variables
  - rather than just passive observation




**NB.** see Judea Pearl's book "Causality"

# The Do Calculus: Interpretation

- for variable $X$, $do(X)$ represents the statement that $X$ has been set by (outside) intervention
- we seek to evaluate and simplify statements involving
  - $A \perp\!\!\!\perp B \mid do(X), Y$ (independence)
  - $p(A \mid B, do(X))$ (probability)
- given a graphical model $G$ over variables $U$, interpret the graph as causal, and the individual node probabilities $p(u|par(u))$ as causal prescriptions

# The Do Calculus: Interpretation, cont.

**Probability theory:** Given a graphical model $G$ over variables $U$, consider $p(A|B)$. Let $anc(V)$ be the ancestral set of $V$, then

$$p(A|B) = \frac{1}{Z} \sum_{anc(A \cup B) \setminus A \cup B} \left( \prod_{u \in anc(A \cup B)} p(u|par(u)) \right)$$

**Causal probability theory:** Given a graphical model $G$ over variables $U$, consider $p(A|B, do(X))$ for $X \subseteq B$. Then

$$p(A|B, do(X)) = \frac{1}{Z} \sum_{anc(A \cup B) \setminus A \cup B} \left( \prod_{u \in anc(A \cup B) \setminus X} p(u|par(u)) \right)$$

$\Longrightarrow$ you eliminate terms $\prod_{x \in X} p(x|par(x))$ !

# The Do Calculus: Interpretation, cont.

**Probability theory:** Given a graphical model $G$ over variables $U$, consider $p(A|B)$. Let $anc(V)$ be the ancestral set of $V$, then

$$p(A|B) = \frac{1}{Z} \sum_{anc(A \cup B) \setminus A \cup B} \left( \prod_{u \in anc(A \cup B)} p(u|par(u)) \right)$$

**Causal probability theory:** Given a graphical model $G$ over variables $U$, consider $p(A|B, do(X))$ for $X \subseteq B$. Then

$$p(A|B, do(X)) = \frac{1}{Z} \sum_{anc(A \cup B) \setminus A \cup B} \left( \prod_{u \in anc(A \cup B) \setminus X} p(u|par(u)) \right)$$

$\Longrightarrow$ you eliminate terms $\prod_{x \in X} p(x|par(x))$ !

$\Longrightarrow$ Pearl's 3 rules of do-calculus follow somewhat simply

# Using Causality

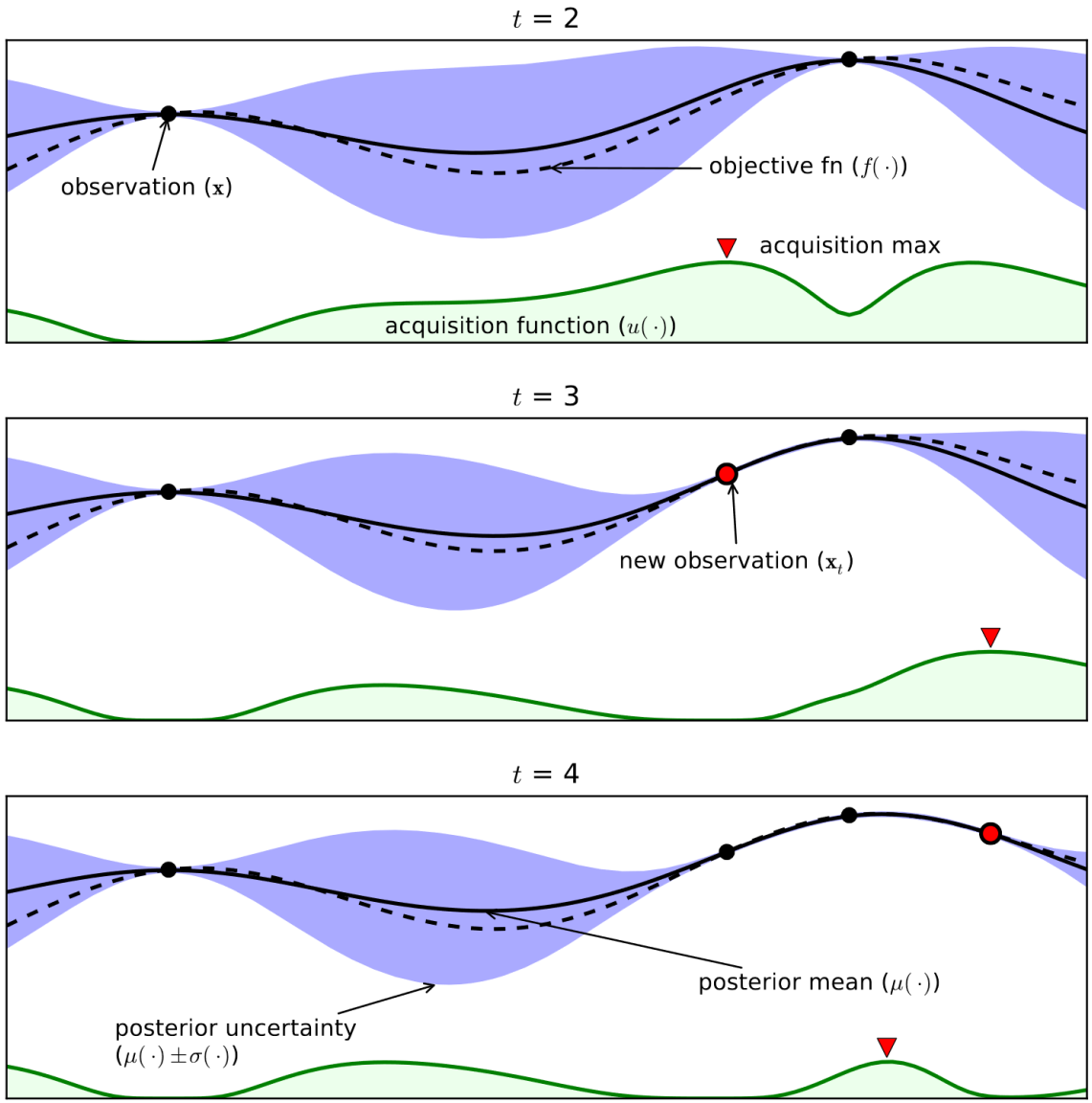
- Simple classification/regression tasks are usually configured by experts:
  - they use their own knowledge of causality to decide what variables to use/collect
- We're now developing:
  - life-long learning
  - multi-task learning

  where hand-configuring by human experts cannot be done.
- More complex learning systems need to "self-configure" so they will need to use causality.

# Bayes-Optimal Parameter Search

- Gaussian processes are a suped-up version of least squares regression that allows:
  - fitting non-linear functions
  - point-wise estimates of uncertainty
- But they only work well in low dimensions.
  - less than 6, so in $\mathcal{R}^6$
- By adding a exploration-exploitation affect, you can use them to do "optimal non-linear optimisation".

# Bayes-Optimal Optimisation

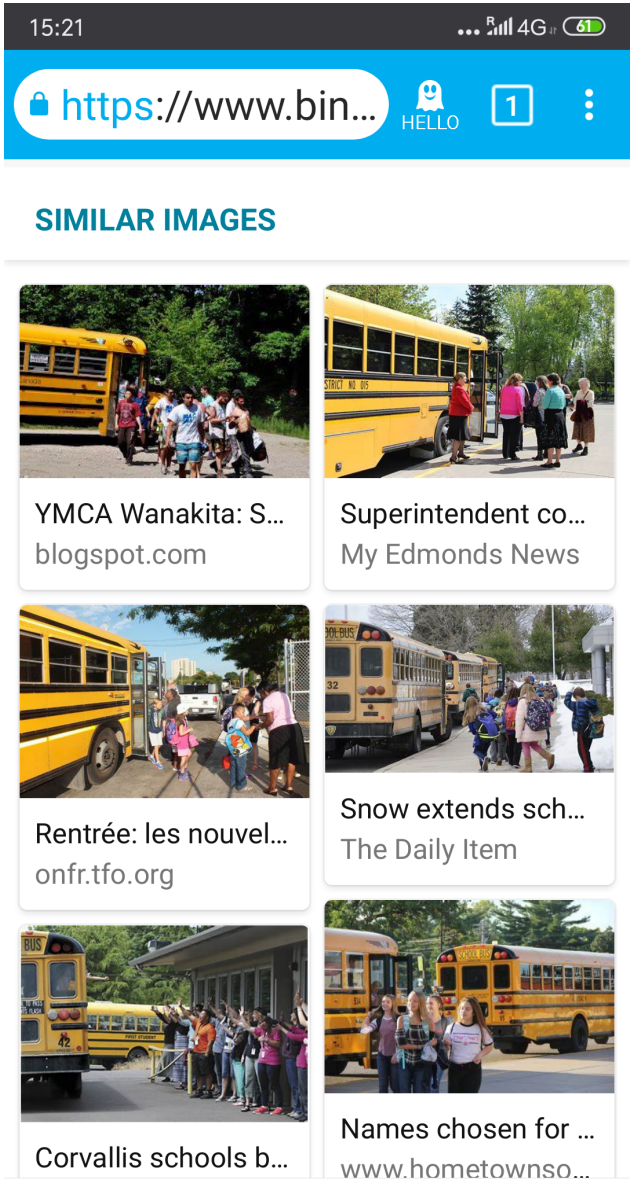


# Hyper-parameter Tuning

- Many learning algorithms have several hyper-parameters
  - $\lambda$ tradeoff in regularisation
  - tree depth in XGBoost
- How do we set them?
- Use Gaussian process optimisation!
- Good packages in Python etc., exist

# Information Retrieval: Images

image search gives

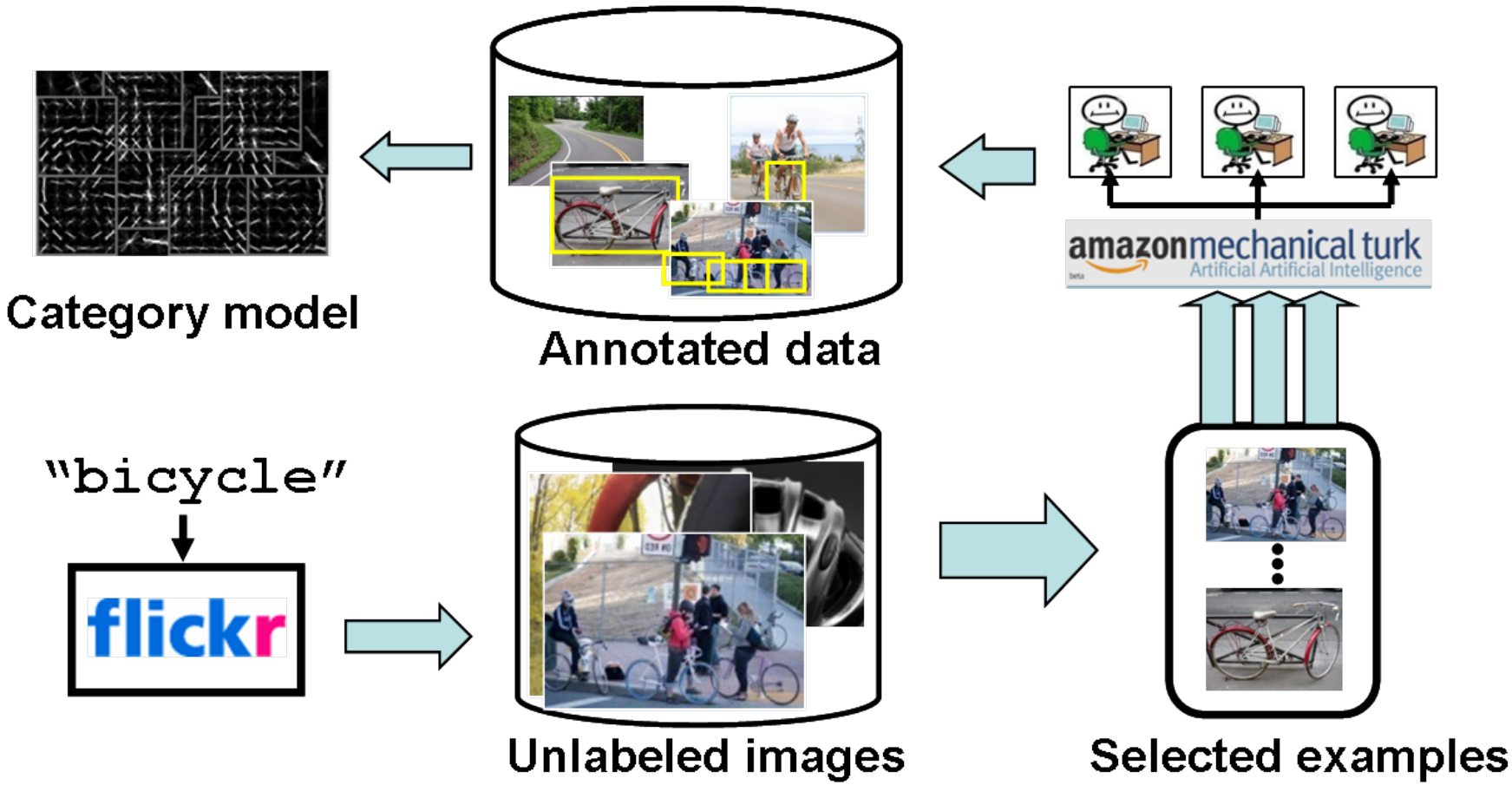


# Information Retrieval becomes ML

Information Retrieval (IR): from a single query (image, text), retrieve "best" matching documents. e.g. text/image search engine

- this is like learning from just one example!
- called zero/one-shot learning in Deep Neural Networks, especially for images
- start of the-art in IR is deep neural networks!
  - see SIGIR 2018 tutorial by Xu, He and Li
  - to my knowledge IR techniques and one-shot researchers do not intersect!
- Google has now installed BERT in their search.
  - is a (text) counterpart technology for one-shot learning

# Active Learning




from `kisspng.com` "active learning machine learning"

WARNING: active learning is an old field, very fragmented, and no good recent survey articles exist.

# Bayesian Theory TL;DR (skip)

- there are strong theoretical justifications for using Bayesian methods
  - betting arguments: Dutch books
  - decision theory arguments: von Neumann-Morgenstern Axioms
  - coherence arguments: Cox's axioms
- but these ignore three key issues:
  - computational complexity
  - multi-agent reasoning
  - model adequacy (the "truth" must be in the model family)
- of the kind that applied ML people well understand!

# Bayesian Theory TL;DR (skip)

- there are strong theoretical justifications for using Bayesian methods
  - betting arguments: Dutch books
  - decision theory arguments: von Neumann-Morgenstern Axioms
  - coherence arguments: Cox's axioms
- but these ignore three key issues:
  - computational complexity
  - multi-agent reasoning
  - model adequacy (the "truth" must be in the model family)
- of the kind that applied ML people well understand!
- and they don't say you must *use* Bayesian methods
  - just that you should be reasonable consistent in results with Bayesian methods

# Applied Bayesian Methods (skip)

- in practice we do hybrids, compromises, and variations, i.e., "applied Bayesian"
  - like regularisation
  - like ensembling
  - so-called "empirical Bayes" (e.g., with cross validation)
  - MDL and MML approaches
- while being cognizant of the three key issues

# Applied Bayesian Methods (skip)

- in practice we do hybrids, compromises, and variations, i.e., "applied Bayesian"
  - like regularisation
  - like ensembling
  - so-called "empirical Bayes" (e.g., with cross validation)
  - MDL and MML approaches
- while being cognizant of the three key issues

Bayesian methods are not an application technology.

They motivate new methods and give alternative intepretations of new methods developed otherwise.

# Metrics and Divergences (skip)

- Often times learning is represented as minimising a cost, sometimes between the model and the empirical data distribution.
  - Given a data set $\{\vec{x}_1, \ldots, \vec{x}_N\}$:

$$p_{data}(\vec{x}) = \frac{1}{N} \sum_{i=1}^{N} 1_{\vec{x}_i = \vec{x}}$$

- To this a regularisation term may be added (to be pseudo-Bayesian).
- So we seek to minimise a cost between a parameterised probabilistic model $p(\vec{x}|\mathbf{\Phi})$ and the empirical data distribution

$$L\left(p_{data}(\vec{x}), p(\vec{x}|\mathbf{\Phi})\right)$$

- At the very least, $L(\cdot, \cdot)$ should be a proper scoring rule.
- To this a regularisation term may be added (to be pseudo-Bayesian).

$$R(\mathbf{\Phi}) = ||\mathbf{\Phi}||^2$$

# Metrics and Divergences, Base Cases

K-L divergence: though sometimes the more complex Jensen-Shannon divergence is used

Total variation: (TV) in discrete domain $X$, is

$$TV(p, q) = \max_{x \in X} |p(x) - q(x)|$$

Others: mean square error, absolute error, ...

# Metrics and Divergences, Base Cases

K-L divergence: though sometimes the more complex Jensen-Shannon divergence is used

Total variation: (TV) in discrete domain $X$, is

$$TV(p, q) = \max_{x \in X} |p(x) - q(x)|$$

Others: mean square error, absolute error, ...

Example cost function:

$$KL\left(p_{data}(\vec{x}), p(\vec{x}|\mathbf{\Phi})\right) + \lambda ||\mathbf{\Phi}||^2$$

# f-Divergence (skip)

Is the class defined on probability functions $p(\cdot)$ and $q(\cdot)$ by

$$\begin{aligned} D_f(p,q) &= \int_x q(x)\, f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x \ , \\ &= \sup_{T(\cdot)} \left(\mathrm{E}_{x\sim p}\big[T(x)\big] - \mathrm{E}_{x\sim p}\big[f^*\big(T(x)\big)\big]\right) \end{aligned}$$

for convex univariate function $f(\cdot)$, its conjugate $f^*(\cdot)$ and $T(\cdot)$ is any function suitable for the domain of $f^*(\cdot)$.

# f-Divergence (skip)

Is the class defined on probability functions $p(\cdot)$ and $q(\cdot)$ by

$$\begin{aligned} D_f(p,q) &= \int_x q(x)\, f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x \ , \\ &= \sup_{T(\cdot)} \left(\mathrm{E}_{x\sim p}\big[T(x)\big] - \mathrm{E}_{x\sim p}\big[f^*\big(T(x)\big)\big]\right) \end{aligned}$$

for convex univariate function $f(\cdot)$, its conjugate $f^*(\cdot)$ and $T(\cdot)$ is any function suitable for the domain of $f^*(\cdot)$.

- second line (conjugate formulation) due to Nguyen, Wainwright, Jordan, 2010

# f-Divergence (skip)

Is the class defined on probability functions $p(\cdot)$ and $q(\cdot)$ by

$$\begin{aligned} D_f(p,q) &= \int_x q(x)\, f\left(\frac{p(x)}{q(x)}\right) \mathrm{d}x \ , \\ &= \sup_{T(\cdot)} \left(\mathrm{E}_{x\sim p}\big[T(x)\big] - \mathrm{E}_{x\sim p}\big[f^*\big(T(x)\big)\big]\right) \end{aligned}$$

for convex univariate function $f(\cdot)$, its conjugate $f^*(\cdot)$ and $T(\cdot)$ is any function suitable for the domain of $f^*(\cdot)$.

- second line (conjugate formulation) due to Nguyen, Wainwright, Jordan, 2010
- generally requires $p(\cdot)$ and $q(\cdot)$ have the same domain (are zero for the same arguments) $\longrightarrow$ models must fit truth!
- includes both K-L and TV as special cases, and also J-S divergence (symmetrised K-L)

# f-Divergence: Examples

| Name | $D_f(P\|Q)$ | Generator $f(u)$ |
|---|---|---|
| Total variation | $\frac{1}{2}\int \lvert p(x)-q(x)\rvert \,\mathrm{d}x$ | $\frac{1}{2}\lvert u-1\rvert$ |
| Kullback-Leibler | $\int p(x)\log\frac{p(x)}{q(x)}\,\mathrm{d}x$ | $u\log u$ |
| Reverse Kullback-Leibler | $\int q(x)\log\frac{q(x)}{p(x)}\,\mathrm{d}x$ | $-\log u$ |
| Pearson $\chi^2$ | $\int \frac{(q(x)-p(x))^2}{p(x)}\,\mathrm{d}x$ | $(u-1)^2$ |
| Neyman $\chi^2$ | $\int \frac{(p(x)-q(x))^2}{q(x)}\,\mathrm{d}x$ | $\frac{(1-u)^2}{u}$ |
| Squared Hellinger | $\int \left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2 \mathrm{d}x$ | $(\sqrt{u}-1)^2$ |
| Jeffrey | $\int (p(x)-q(x))\log\left(\frac{p(x)}{q(x)}\right)\mathrm{d}x$ | $(u-1)\log u$ |
| Jensen-Shannon | $\frac{1}{2}\int p(x)\log\frac{2p(x)}{p(x)+q(x)} + q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x$ | $-(u+1)\log\frac{1+u}{2} + u\log u$ |
| Jensen-Shannon-weighted | $\int p(x)\pi\log\frac{p(x)}{\pi p(x)+(1-\pi)q(x)} + (1-\pi)q(x)\log\frac{q(x)}{\pi p(x)+(1-\pi)q(x)}\,\mathrm{d}x$ | $\pi u\log u - (1-\pi+\pi u)\log(1-\pi+\pi u)$ |
| GAN | $\int p(x)\log\frac{2p(x)}{p(x)+q(x)} + q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x - \log(4)$ | $u\log u - (u+1)\log(u+1)$ |
| $\alpha$-divergence ($\alpha\notin\{0,1\}$) | $\frac{1}{\alpha(\alpha-1)}\int \left(p(x)\left[\left(\frac{q(x)}{p(x)}\right)^{\alpha}-1\right]-\alpha(q(x)-p(x))\right)\mathrm{d}x$ | $\frac{1}{\alpha(\alpha-1)}\left(u^{\alpha}-1-\alpha(u-1)\right)$ |

from Nowizin's GAN tutorial, MLSS 2018, Madrid

Most metrics have large tables of alternatives like this.

# Bregman Divergence (skip)

Is the class defined on discrete probability vectors $\vec{p}$ and $\vec{q}$ as

$$B_F(\vec{p}, \vec{q}) = F(\vec{p}) - F(\vec{q}) - (\vec{p} - \vec{q})^T \nabla F(\vec{q})$$

for convex vector function $F(\cdot)$.

- includes K-L and a variation gives TV
- intersection with f-divergence is the class of alpha-divergences or Rényi entropies

# Transport Metrics (skip)

Is the class defined on probability functions $p(\cdot)$ and $q(\cdot)$, given distance $d(\cdot, \cdot)$ sharing the same domain

$$W_{d(\cdot,\cdot)}(p, q) = \sup_{T(\cdot)} \left| \mathrm{E}_{x \sim p}\left[T(x)\right] - \mathrm{E}_{x \sim q}\left[T(x)\right] \right|$$

where $T(\cdot)$ must satisfy $|T(x) - T(y)| \leq d(x, y)$.

- for standard Euclidean metric, this means $T(\cdot)$ is 1-Lipschitz
  - coarsely implemented with deep networks using "weight clipping"
- includes TV as the trivial case where $d(x, y) = 1_{x=y}$
- rather similar in form to second variation of f-divergence (indeed, they are the same for TV)

# Metrics and Divergences, cont.

- An embarrassment of riches!
- But what guidance do we have regards the best choice?
  - theory is more about relationships and general properties

# Metrics and Divergences, cont.

- An embarrassment of riches!
- But what guidance do we have regards the best choice?
  - theory is more about relationships and general properties
- K-L divergence where the first argument is the empirical data distribution is equivalent to maximum likelihood.
  - an old result from the physicists

# Metrics and Divergences, cont.

- An embarrassment of riches!
- But what guidance do we have regards the best choice?
  - theory is more about relationships and general properties
- K-L divergence where the first argument is the empirical data distribution is equivalent to maximum likelihood.
  - an old result from the physicists
- Reasonable criteria, based on large sample theory, is they be proper scoring rules.
  - most differentiable cost functions are, e.g., not TV!

# Metrics and Divergences, cont.

- An embarrassment of riches!
- But what guidance do we have regards the best choice?
  - theory is more about relationships and general properties
- K-L divergence where the first argument is the empirical data distribution is equivalent to maximum likelihood.
  - an old result from the physicists
- Reasonable criteria, based on large sample theory, is they be proper scoring rules.
  - most differentiable cost functions are, e.g., not TV!
- Far less guidance in the small sample case.
  - Bayesian ideas can help!

# Metrics and Divergences, cont.

- An embarrassment of riches!
- But what guidance do we have regards the best choice?
  - theory is more about relationships and general properties
- K-L divergence where the first argument is the empirical data distribution is equivalent to maximum likelihood.
  - an old result from the physicists
- Reasonable criteria, based on large sample theory, is they be proper scoring rules.
  - most differentiable cost functions are, e.g., not TV!
- Far less guidance in the small sample case.
  - Bayesian ideas can help!
- Some applications are not model fitting but data summarisation.

  e.g., hierarchical clustering
  - so motivation for selection of cost is completely different

# The K-L Variational Setup (skip)

The variational method, also called mean field is based on the following information theoretic identity:

$$\log p(\boldsymbol{x}|\Theta) - KL\left(q(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}|\boldsymbol{x},\Theta)\right) \quad (1)$$

$$= \quad \mathrm{E}_{q()}\left[\log p(\boldsymbol{x},\boldsymbol{z}|\Theta)\right] + H(q()) \quad (2)$$

- data is $\boldsymbol{x}$, latent variables $\boldsymbol{z}$, model parameters $\Theta$
- $q(\boldsymbol{z}|\boldsymbol{x})$ is an introduced probability density and we can vary how we like
- theoreticians have tried to reproduce/generalise this to other divergence/metric paradigms
  - haven't succeeded
  - the relationships between probability, independence and K-L are unique!

# The K-L Variational Setup (skip)

The variational identity:

$$\log p(\boldsymbol{x}|\Theta) - KL\left(q(\boldsymbol{z}|\boldsymbol{x}) \| p(\boldsymbol{z}|\boldsymbol{x}, \Theta)\right) \quad (1)$$

$$= \quad \mathrm{E}_{q()}\left[\log p(\boldsymbol{x}, \boldsymbol{z}|\Theta)\right] + H(q()) \quad (2)$$

- To maximise $p(\boldsymbol{x}|\Theta)$, therefore repeatedly
  1. increase (1): get $q(\boldsymbol{z}|\boldsymbol{x})$ closer to $p(\boldsymbol{z}|\boldsymbol{x}, \Theta)$ in K-L
  2. increase (2): by optimising/gradient-descent w.r.t. $\Theta$

# The K-L Variational Setup (skip)

The variational identity:

$$\log p(\boldsymbol{x}|\Theta) - KL\left(q(\boldsymbol{z}|\boldsymbol{x})\|p(\boldsymbol{z}|\boldsymbol{x},\Theta)\right) \quad (1)$$

$$= \quad \mathrm{E}_{q()}\left[\log p(\boldsymbol{x},\boldsymbol{z}|\Theta)\right] + H(q()) \quad (2)$$

- To maximise $p(\boldsymbol{x}|\Theta)$, therefore repeatedly
  1. increase (1): get $q(\boldsymbol{z}|\boldsymbol{x})$ closer to $p(\boldsymbol{z}|\boldsymbol{x},\Theta)$ in K-L
  2. increase (2): by optimising/gradient-descent w.r.t. $\Theta$
- for exponential family, Step 1 & 2 are usually simple
- when $q(\boldsymbol{z}|\boldsymbol{x}) = p(\boldsymbol{z}|\boldsymbol{x},\Theta)$, a local maxima is achieved!

# A Probabilistic Model (ML circa 2000s)

1. For each topic $k$:
   (a) For each doc-label $l$: Draw $\lambda_{l,k} \sim \mathrm{Ga}(\mu_0, \mu_0)$
   (b) For each word-feature $l'$: Draw $\delta_{l',k} \sim \mathrm{Ga}(\nu_0, \nu_0)$
   (c) For each token $v$: Compute $\beta_{k,v} = \prod_{l'=1}^{L_{word}} \delta_{l',k}^{g_{v,l'}}$
   (d) Draw $\boldsymbol{\phi}_k \sim \mathrm{Dir}_V(\boldsymbol{\beta}_k)$
2. For each document $d$:
   (a) For each topic $k$: Compute $\alpha_{d,k} = \prod_{l=1}^{L_{doc}} \lambda_{l,k}^{f_{d,l}}$
   (b) Draw $\boldsymbol{\theta}_d \sim \mathrm{Dir}_K(\boldsymbol{\alpha}_d)$
   (c) For each word in document $d$:
      i. Draw topic $z_{d,i} \sim \mathrm{Cat}_K(\boldsymbol{\theta}_d)$
      ii. Draw word $w_{d,i} \sim \mathrm{Cat}_V(\boldsymbol{\phi}_{z_{d,i}})$

from Zhao etal. KAIS 2017

Probabilistic model for fancy topic modelling: MetaLDA.

# A Probabilistic Model, cont.

## Deriving the posterior:

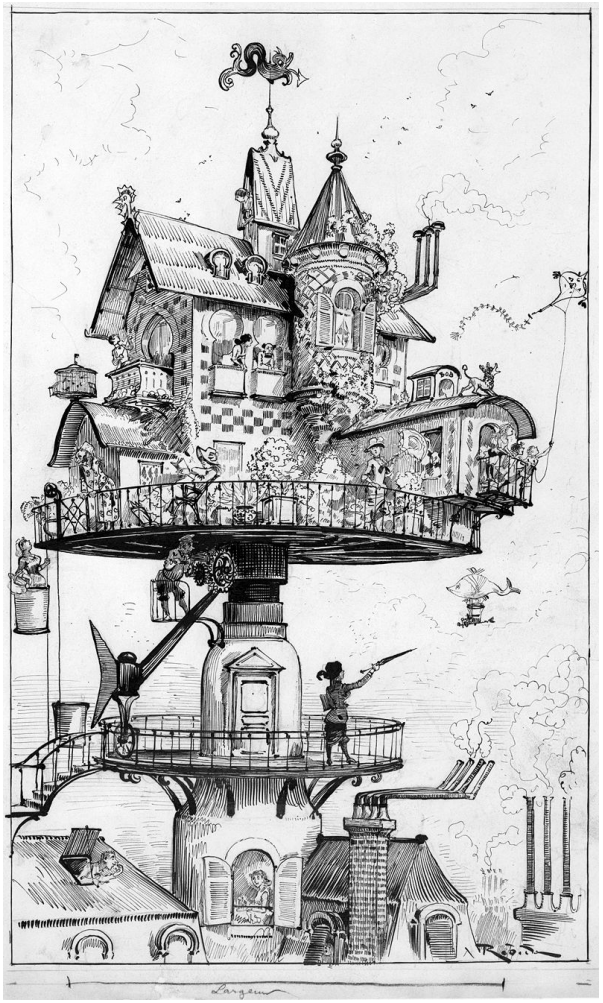
$$\begin{aligned}
&\Pr(\boldsymbol{z}_{1:D}, \boldsymbol{w}_{1:D}; \boldsymbol{\alpha}_{1:D}, \boldsymbol{\beta}_{1:K}) \\
&= \int_{\boldsymbol{\theta}} \prod_{d=1}^{D} \frac{\Gamma(\sum_{k=1}^{K} \alpha_{d,k})}{\prod_{k=1}^{K} \Gamma(\alpha_{d,k})} \prod_{k=1}^{K} \theta_{d,k}^{m_{d,k}+\alpha_{d,k}-1} \int_{\boldsymbol{\phi}} \prod_{k=1}^{K} \frac{\Gamma(\sum_{v=1}^{V} \beta_{k,v})}{\prod_{v=1}^{V} \Gamma(\beta_{k,v})} \prod_{v=1}^{V} \phi_{k,v}^{n_{k,v}+\beta_{k,v}-1}. \\
&= \prod_{d=1}^{D} \frac{\mathrm{Beta}_K(\boldsymbol{\alpha}_d + \boldsymbol{m}_d)}{\mathrm{Beta}_K(\boldsymbol{\alpha}_d)} \prod_{k=1}^{K} \frac{\mathrm{Beta}_V(\boldsymbol{\beta}_k + \boldsymbol{n}_k)}{\mathrm{Beta}_V(\boldsymbol{\beta}_k)} \qquad (2)
\end{aligned}$$

## Deriving a Gibbs sampler:

$$\begin{aligned}
\Pr(z_{d,i} = k \mid \boldsymbol{z}_{1:D}^{-z_{d,i}}, \boldsymbol{w}_{1:D}, \boldsymbol{\alpha}_{1:D}, \boldsymbol{\beta}_{1:K}) &= \frac{\Pr(z_{d,i} = k, \boldsymbol{z}_{1:D}^{-z_{d,i}}, \boldsymbol{w}_{1:D}, \boldsymbol{\alpha}_{1:D}, \boldsymbol{\beta}_{1:K})}{\Pr(\boldsymbol{z}_{1:D}^{-z_{d,i}}, \boldsymbol{w}_{1:D}, \boldsymbol{\alpha}_{1:D}, \boldsymbol{\beta}_{1:K})} \\
&\propto (\alpha_{d,k} + m_{d,k}) \frac{\beta_{k,v} + n_{k,v}}{\beta_{k,\cdot} + n_{k,\cdot}}. \qquad (3)
\end{aligned}$$

A complex task, takes advanced statistical training!

# Outline



# What did We Learn from Old ML?

- models as first-class objects
  - e.g., Bayesian & Markov networks, neural networks
- how to do linear and partitioning models well
  - e.g., XGBoost, online LDA, exponential family, additive models
- how to augment them with fancy mathematical tricks
  - e.g., kernels, non-parametrics (infinite vectors)
- cost functions and statistical ML theory
  - e.g., bias-variance tradeoff, regularisation, metrics and divergences, variational methods
- paradigms
  - e.g., online learning, active learning, transfer learning

Bengio et al.s' "Representation Learning: A Review and New Perspectives" 2013 *IEEE PAMI* article has a nice summary!

# How Does Deep Learning Innovate?

- Model/Spec driven black-box algorithms ease the workload of developers.
  - machine learning without statistics!

# How Does Deep Learning Innovate?

- Model/Spec driven black-box algorithms ease the workload of developers.
  - machine learning without statistics!
- Porting down to GPUs or multi-core allows real speed.

# How Does Deep Learning Innovate?

- Model/Spec driven black-box algorithms ease the workload of developers.
  - machine learning without statistics!
- Porting down to GPUs or multi-core allows real speed.
- Learning representations and discovering higher order concepts.
  - convolutions, structures, sequences, ...

# How Does Deep Learning Innovate?

- Model/Spec driven black-box algorithms ease the workload of developers.
  - machine learning without statistics!
- Porting down to GPUs or multi-core allows real speed.
- Learning representations and discovering higher order concepts.
  - convolutions, structures, sequences, ...
- High capacity makes them very flexible in fitting and does implicit parallel search.

# How Does Deep Learning Innovate?

- Model/Spec driven black-box algorithms ease the workload of developers.
  - machine learning without statistics!
- Porting down to GPUs or multi-core allows real speed.
- Learning representations and discovering higher order concepts.
  - convolutions, structures, sequences, ...
- High capacity makes them very flexible in fitting and does implicit parallel search.
- Allows "modelling in the large":
  - learning to learning
  - multi-task learning
  - imitation learning
  - convolutions, structures, sequences, ...

# How Does Deep Learning Innovate?

- Model/Spec driven black-box algorithms ease the workload of developers.
  - machine learning without statistics!
- Porting down to GPUs or multi-core allows real speed.
- Learning representations and discovering higher order concepts.
  - convolutions, structures, sequences, ...
- High capacity makes them very flexible in fitting and does implicit parallel search.
- Allows "modelling in the large":
  - learning to learning
  - multi-task learning
  - imitation learning
  - convolutions, structures, sequences, ...

  NB. **we can** do some of this with hierarchical Bayesian models

# The Old Versus The New: I

**The Old:** need experts to carefully design algorithms:

- experts need knowledge of distributions and techniques like variational algorithms or Gibbs samplers to construct algorithms
- statistical knowledge intensive

**The New:** (semi) automatic black-box algorithms:

- automatic differentiation, ADAM optimisation, etc.
- port down to GPUs or multi-core, etc.
- easier to scale algorithms

# The Old Versus The New: II

**The Old:** modelling in the small:

- huge range of components can be used
- individual components need care and attention for algorithm development

**The New:** modelling in the large:

- whole blocks can be composed
- general purpose methods deal with it
- restricted in allowable components
  - use concrete distribution and reparameterisation trick

# The Old Versus The New: III

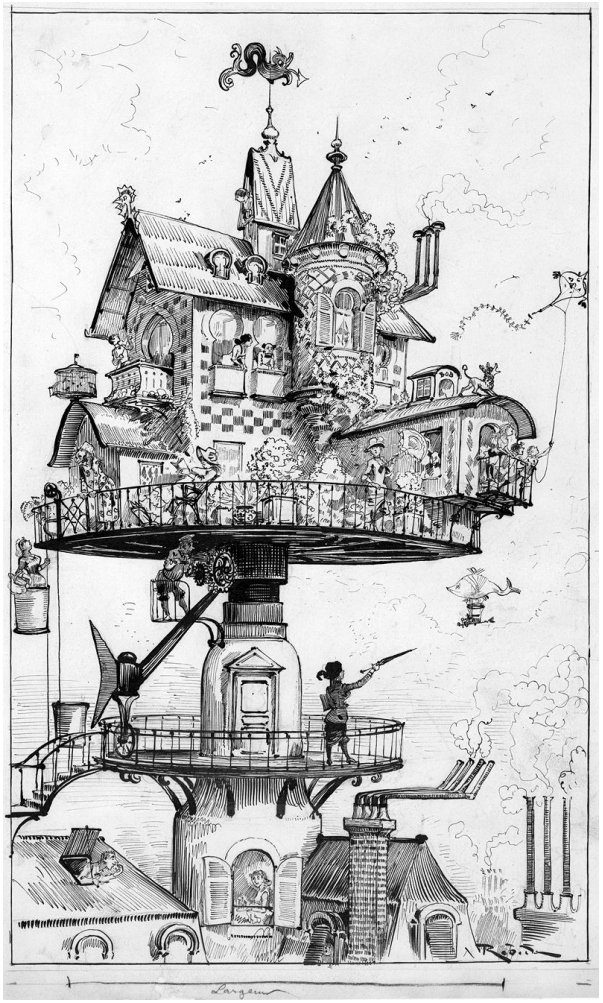
**The Old:** components often directly interpretable:

- parameter vectors can have easy interpretation

**The New:** black-box model requires "explanation" support:

- cannot interpret the model
- need techniques like LIME and SHAP to intepret results

# Outline



# Making Deep Learning Work

**Cost function example:**

$$KL\left(p_{data}(\vec{x}), p(\vec{x}|\mathbf{\Phi})\right) + \lambda||\mathbf{\Phi}||^2$$

- cost function has an error term and a regularisation term
- use a coding language such as TensorFlow to specify an architecture (the model with parameters $\mathbf{\Phi}$)
- automatic differentation can be used to generate code for derivatives
- hooked up to a self-tuning and stochastic variation of gradient descent

# Making Deep Learning Work

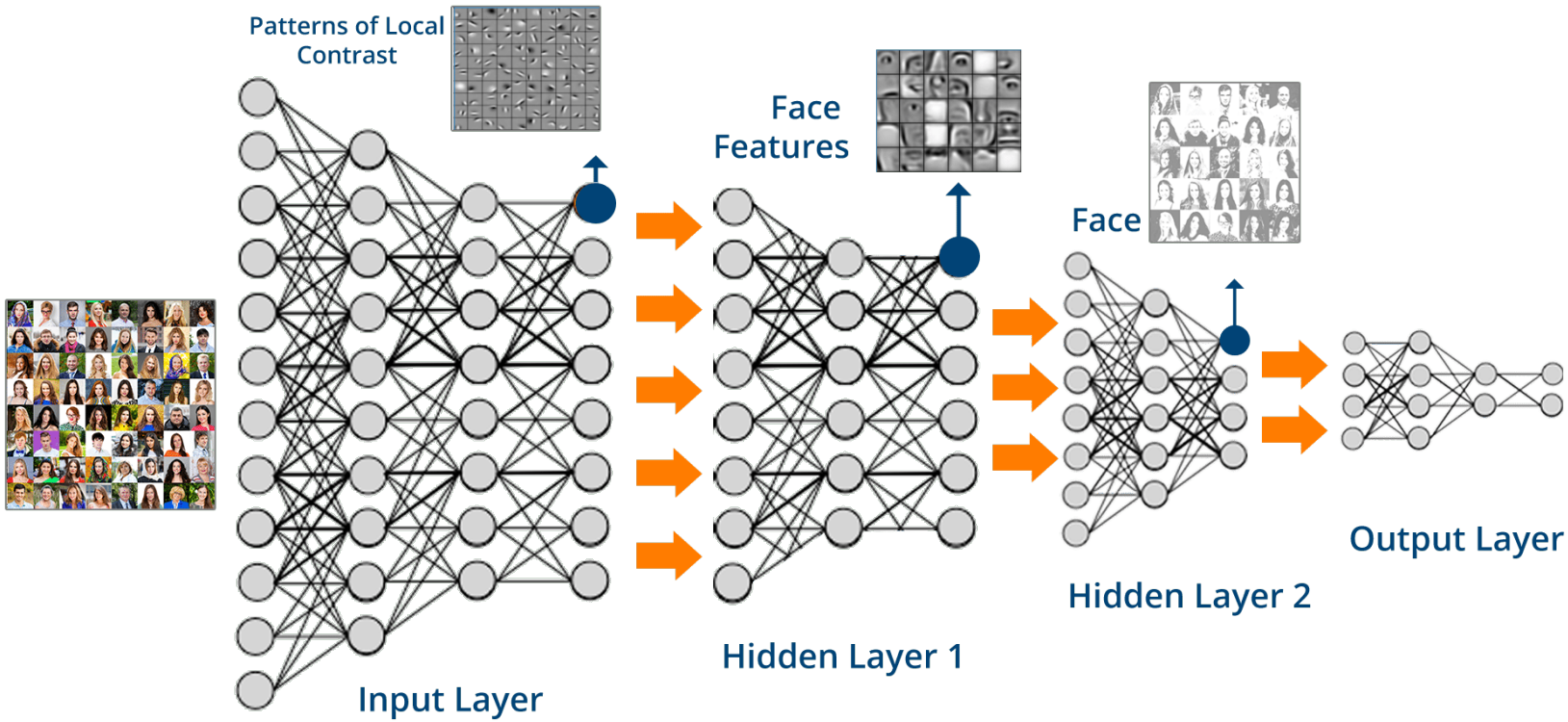
**Cost function example:**

$$KL\left(p_{data}(\vec{x}), p(\vec{x}|\mathbf{\Phi})\right) + \lambda||\mathbf{\Phi}||^2$$

- cost function has an error term and a regularisation term
- use a coding language such as TensorFlow to specify an architecture (the model with parameters $\mathbf{\Phi}$)
- automatic differentation can be used to generate code for derivatives
- hooked up to a self-tuning and stochastic variation of gradient descent

The cost function is approximately minimised semi-automatically, so the effort is in specifying the architecture and the cost function.

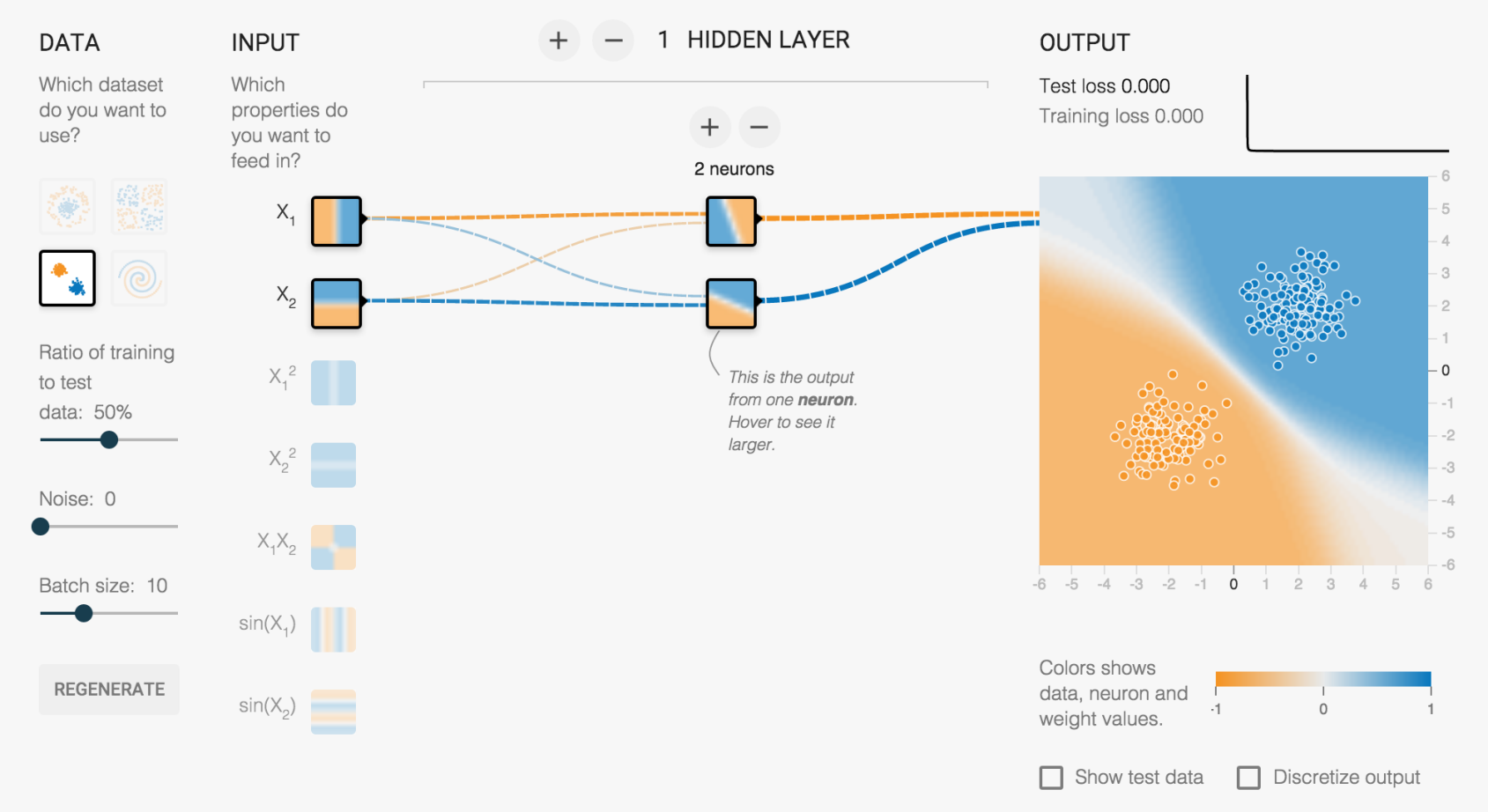
# Deep Representations



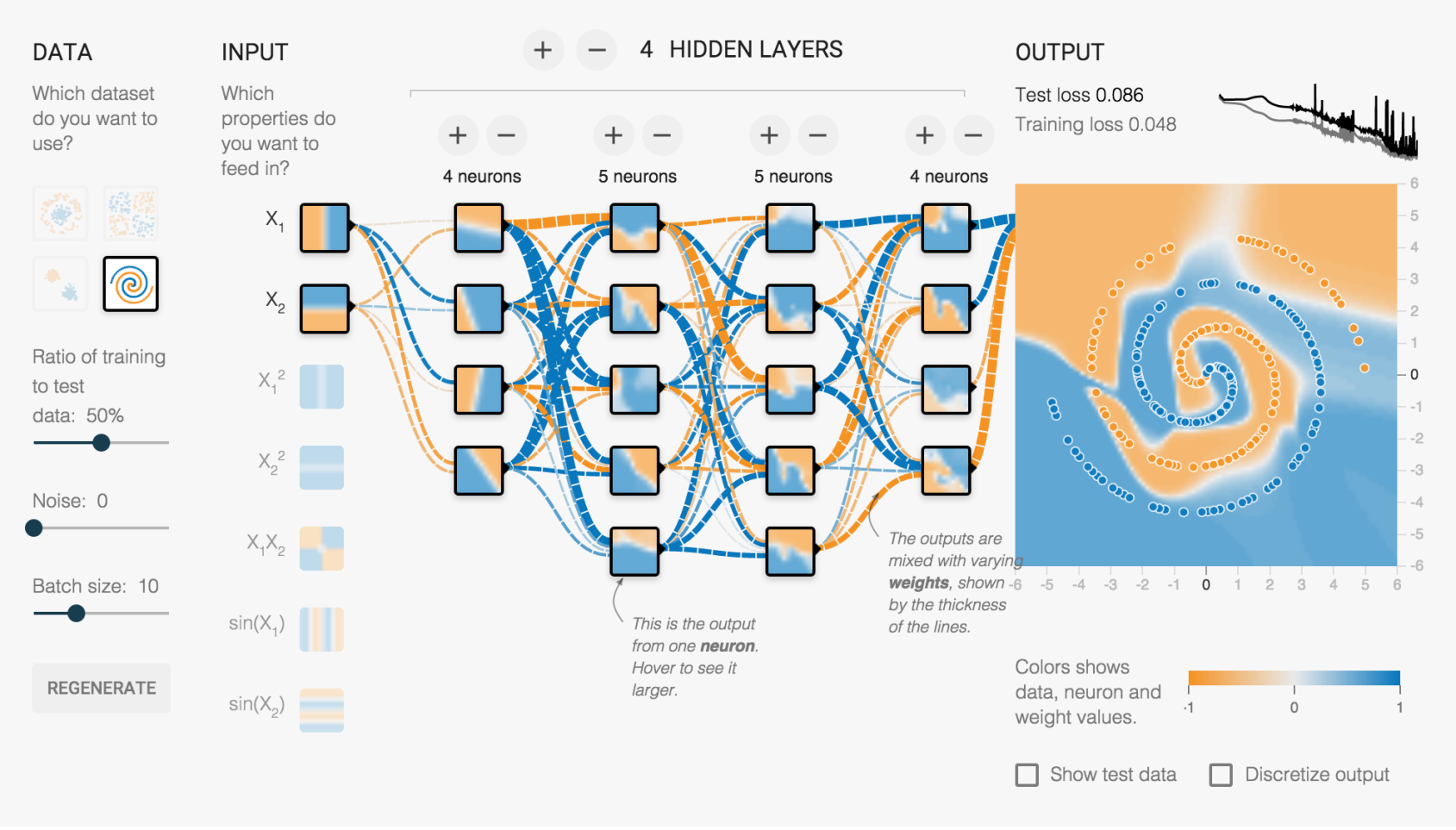

from https://thedatascientist.com/what-deep-learning-is-and-isnt/

**IDEA:** different layers of the network "learn" alternative representations of elements in the image

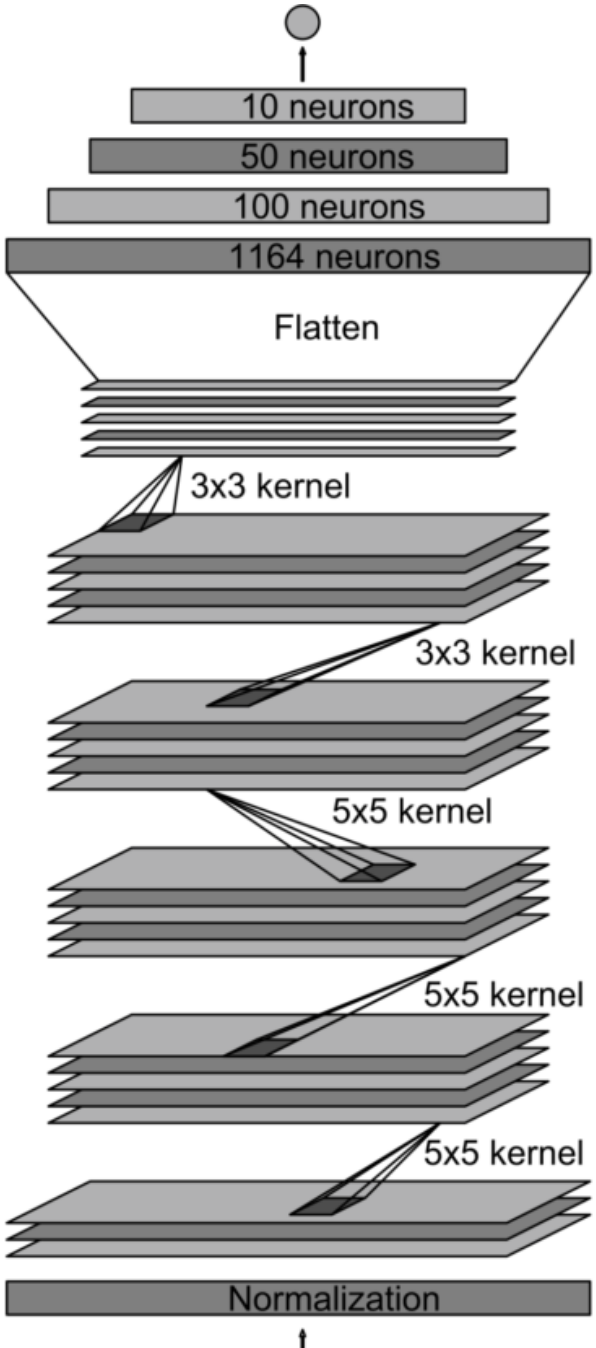
# A Simple Network



# A Bigger Network



# A Real Network from Nvidia




- CNN architecture for driving a car from 2016
- 27M connections but 250k parameters
- convolutions mean there are less parameters than connections
- the working system is compiled from high-level architectural specs

# Symmetric and Sparsity in Modelling

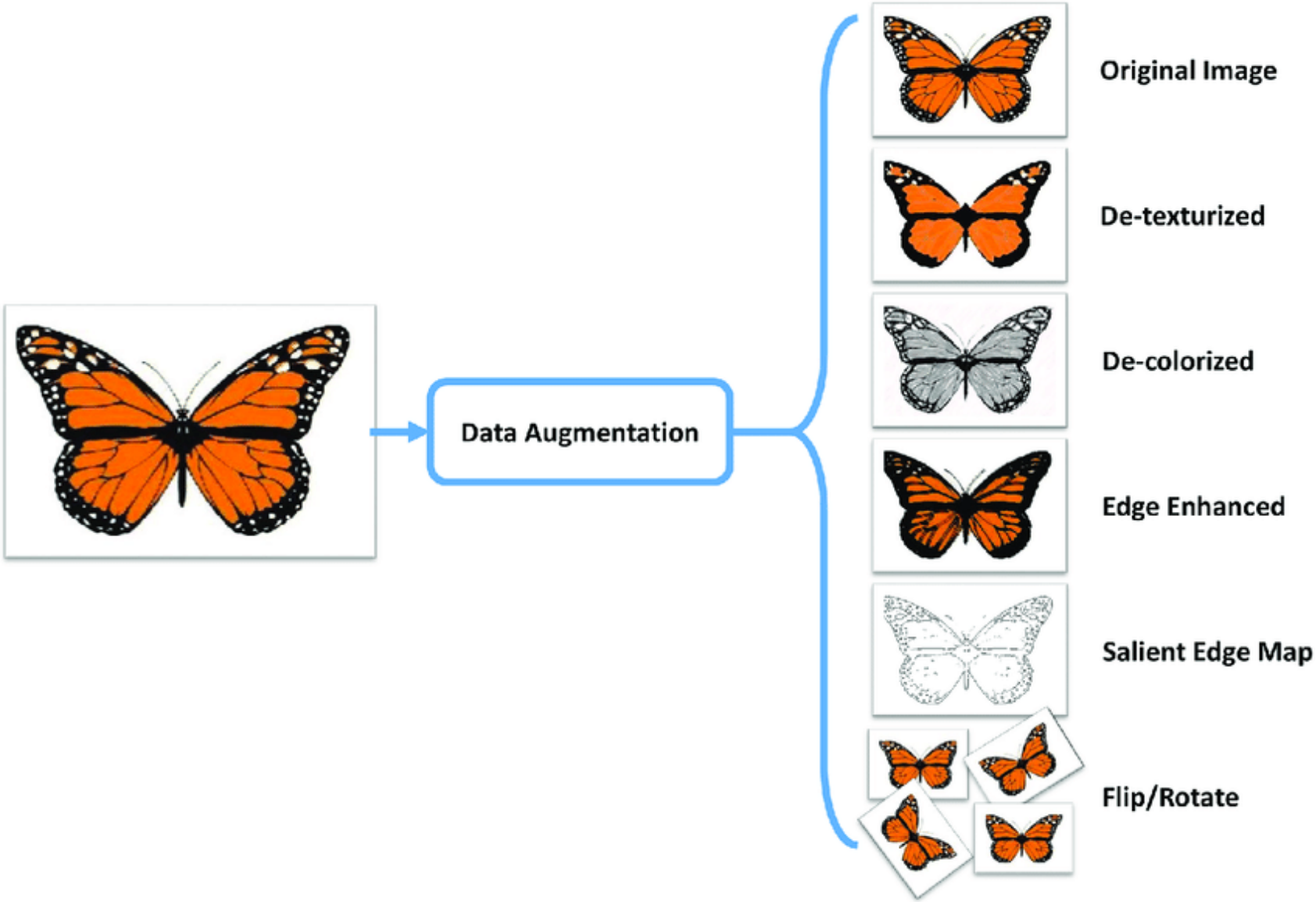
**Observation:** in 2005 I tried adding complex word priors to LDA that had the effect of breaking the symmetry in the model
$\Longrightarrow$ fitting was terrible despite being a good prior!

Model symmetry: identical blocks in a model with random initialisation allow alternatives to appear organically

Model sparsity: many alternatives act as a form of implicit parallelism

These two constructs help explain a lot of the surprising empirical observations where deep NNs seemingly defy statistical learning theory

# Data Augmentation




from Ahmad, Muhammad and Baik, PLoS ONE 2017

# Data Augmentation: MNIST

- technique first used in Zipcode recognition for the US Post
- images can be rotated, shifted, thinned, etc.

# Data Augmentation, MNIST Discussion

- Examples:
  - for digit recognition from images, we want invariance under (1) small shifts, (2) small rotations, (3) thickness changes.
  - for bank loan assessment, we want monotonicity in (1) salary and (2) bank balance, but not age
- It is hard to design these in as invariants of the resultant classification algorithm.
- It is easy to do data augmentation: generate 10 variants of each digit to use for training.
- Recent differential/regularisation techniques can also "encourage" invariants and monotonicity to hold
  - also useful for introducing fairness in learning

# Data Augmentation, Discussion

- We need to identify an invariant in our data we want to hold.

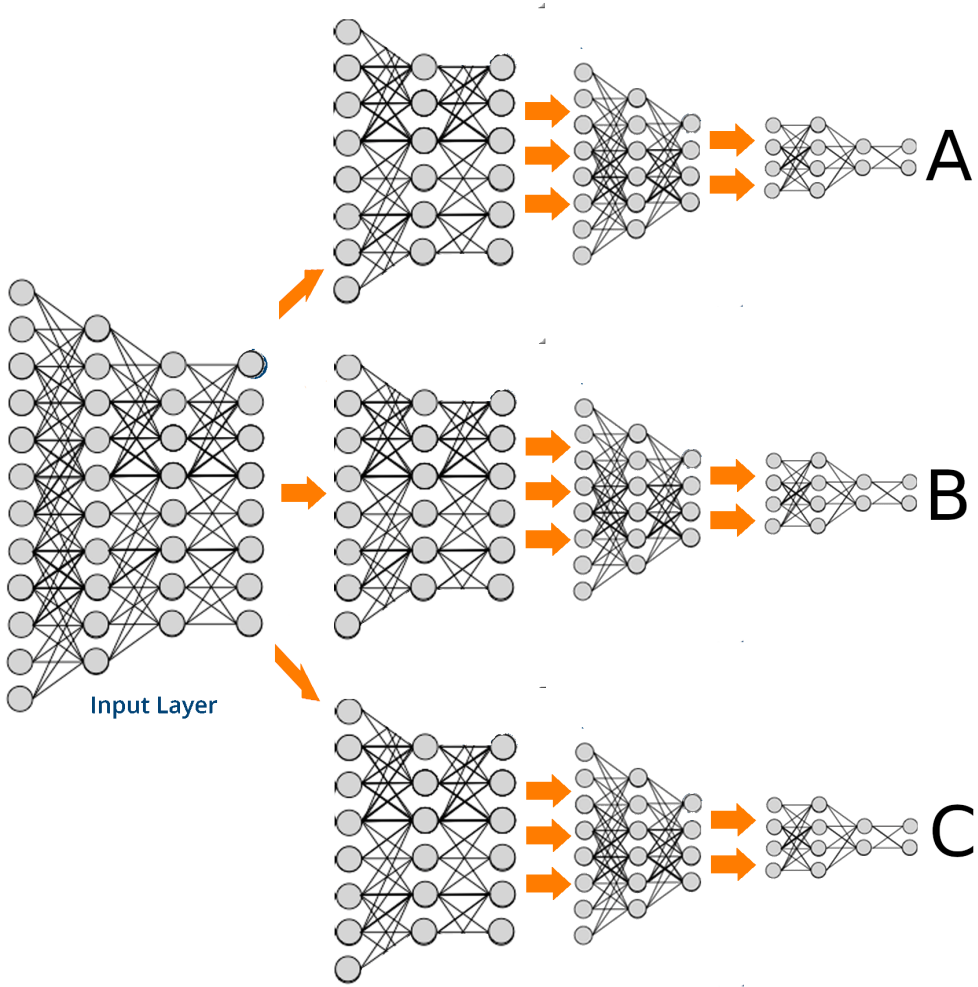
e.g. For text, we could replace synonyms, convert cases (active to passive, past to future tense), etc.

- We need an algorithm to apply changes to the data reflecting the invariant.
- Probabilistic model, for the augmentation distribution Aug, treats it like a convolution:

$$p(y_i \mid x_i, \Theta) \qquad \text{standard data likelihood}$$

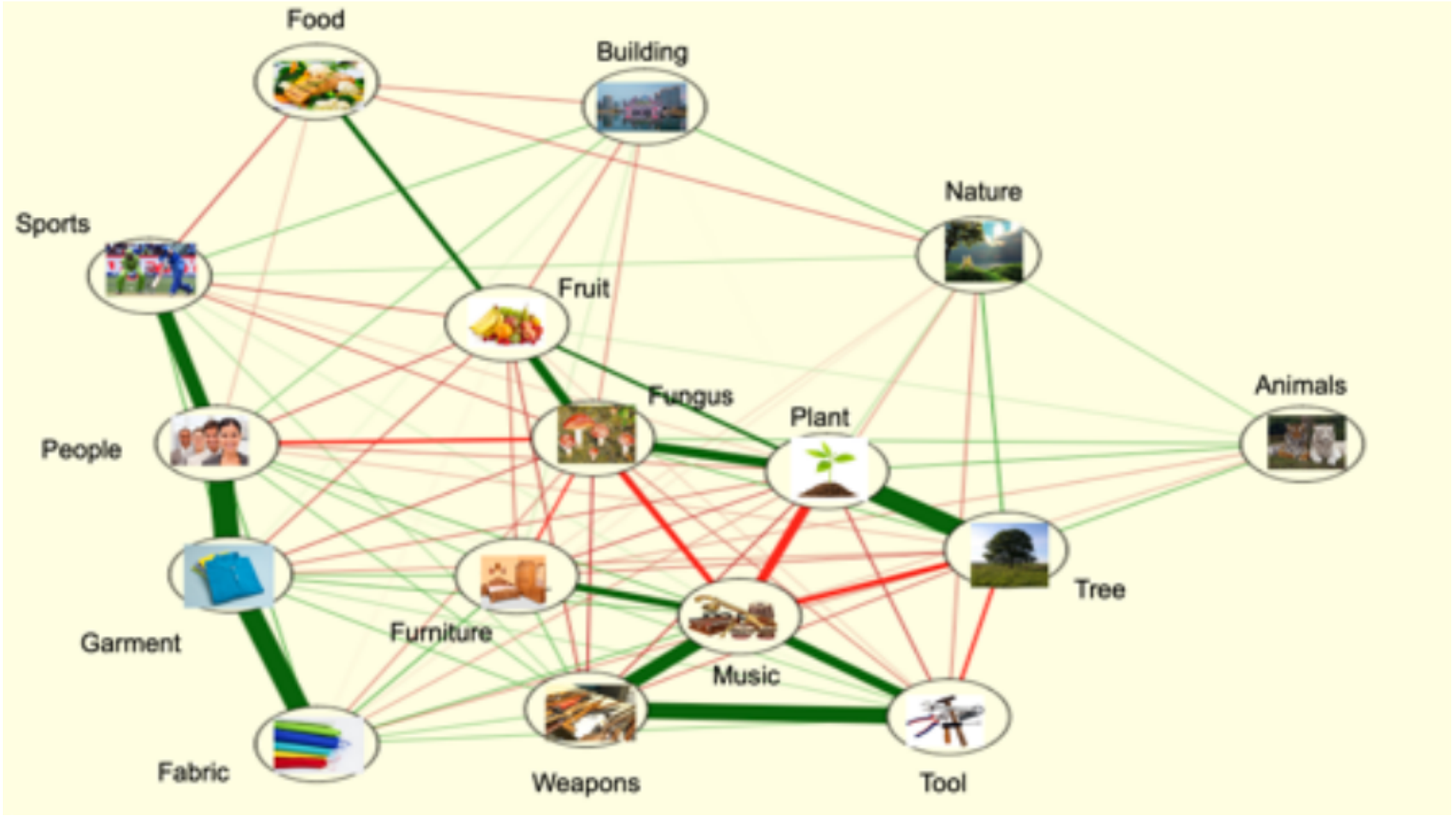
$$\int_{x'} p(y_i \mid x_i', \Theta) p(x' | x_i, \text{Aug}) \, \mathrm{d}x' \qquad \text{augmented data likelihood}$$

# Multi-Task Learning: Sharing




- 3 tasks A, B and C
- share lower layers of the network as “pattern” features
- upper layers specialise for tasks

# Feature Sharing

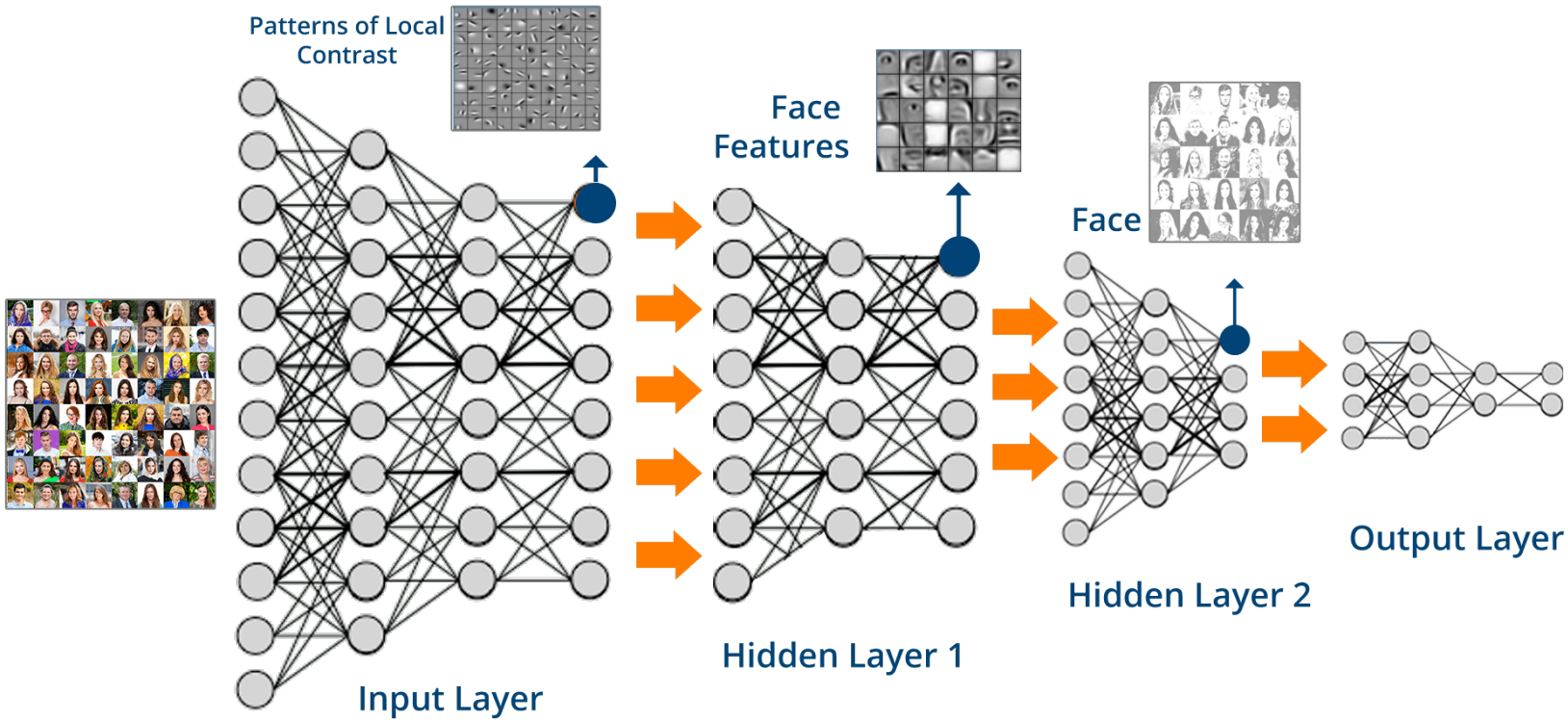



figure

from Bhattacharjee et al, ArXiv, 2019

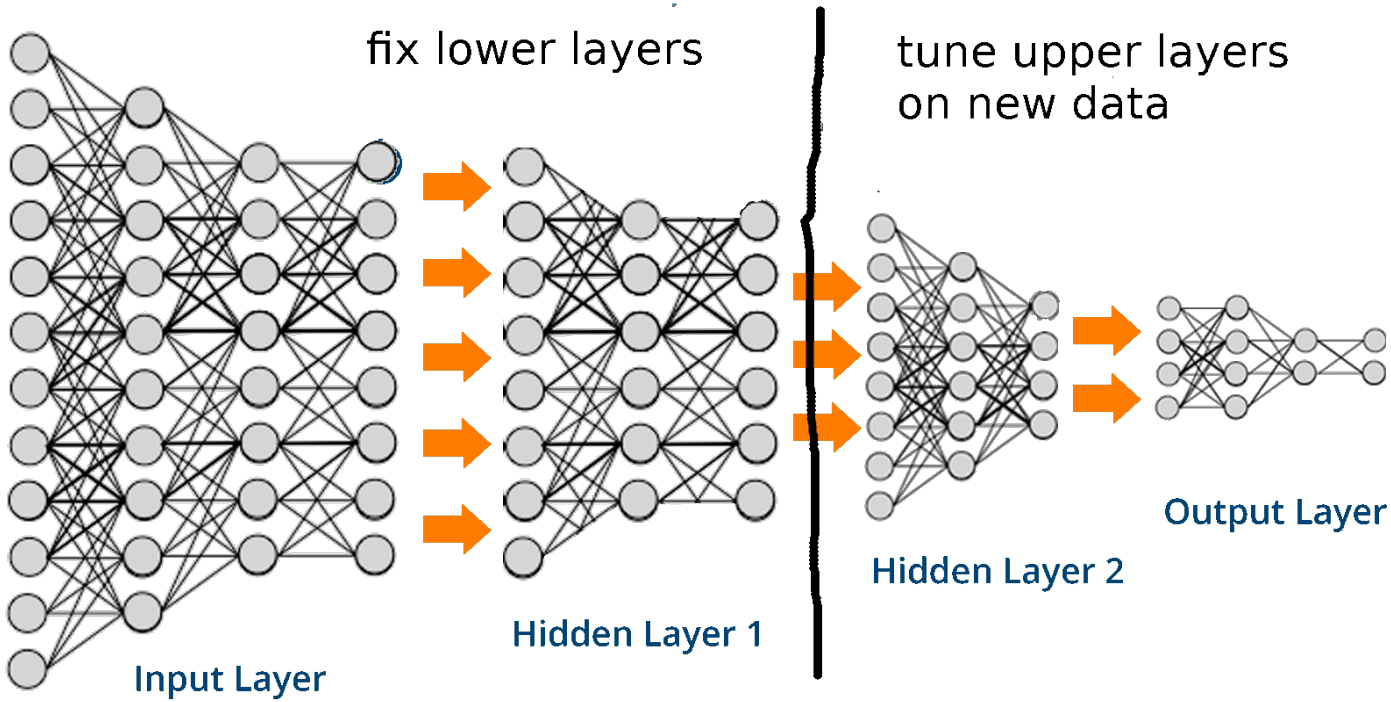
Shows amount of feature sharing done by a multi-task object categorisation system. In this case, level of sharing was learnt.

# Pre-Training: Initialisation



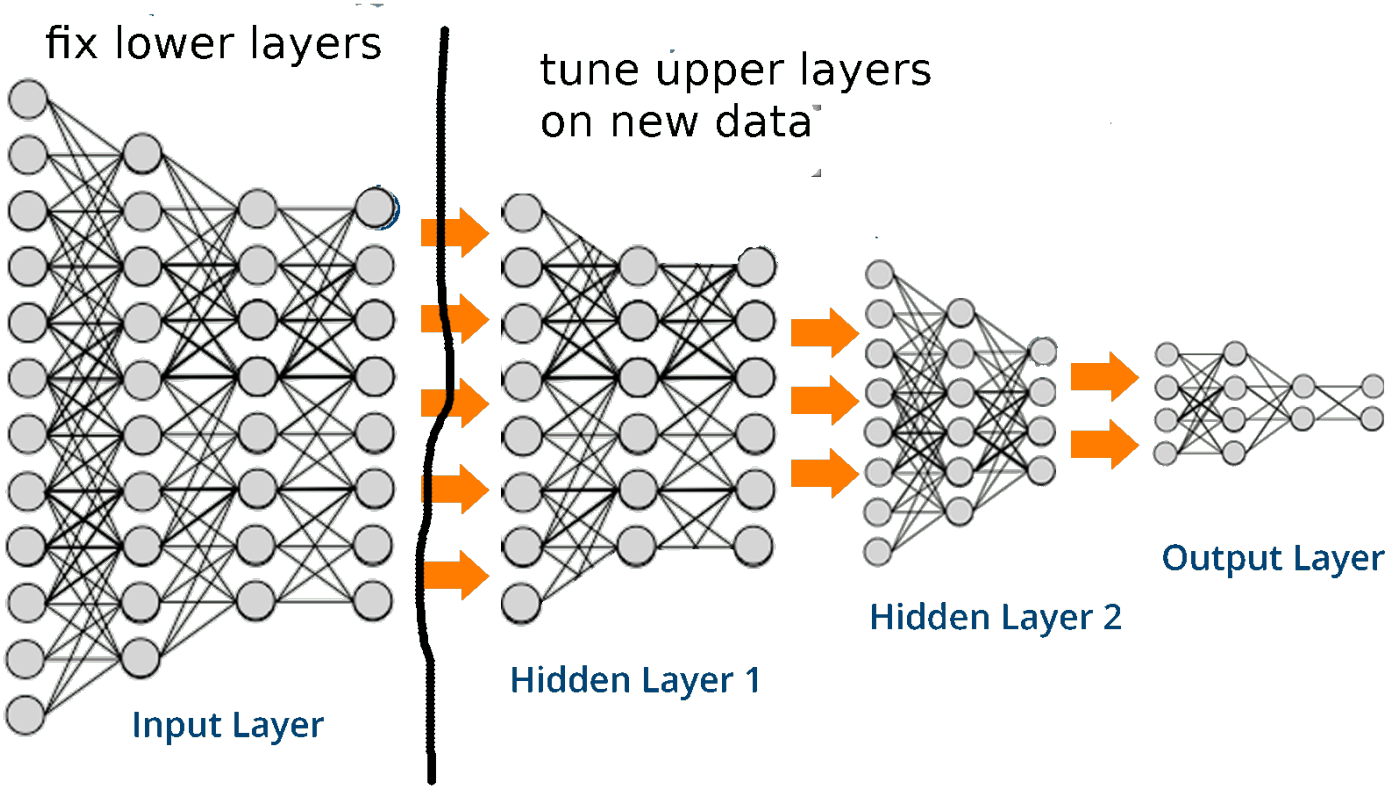

1. train the network initially on a very large set of "somewhat" related images
2. leave the values as an initialisation

# Pre-Training: Fix Features



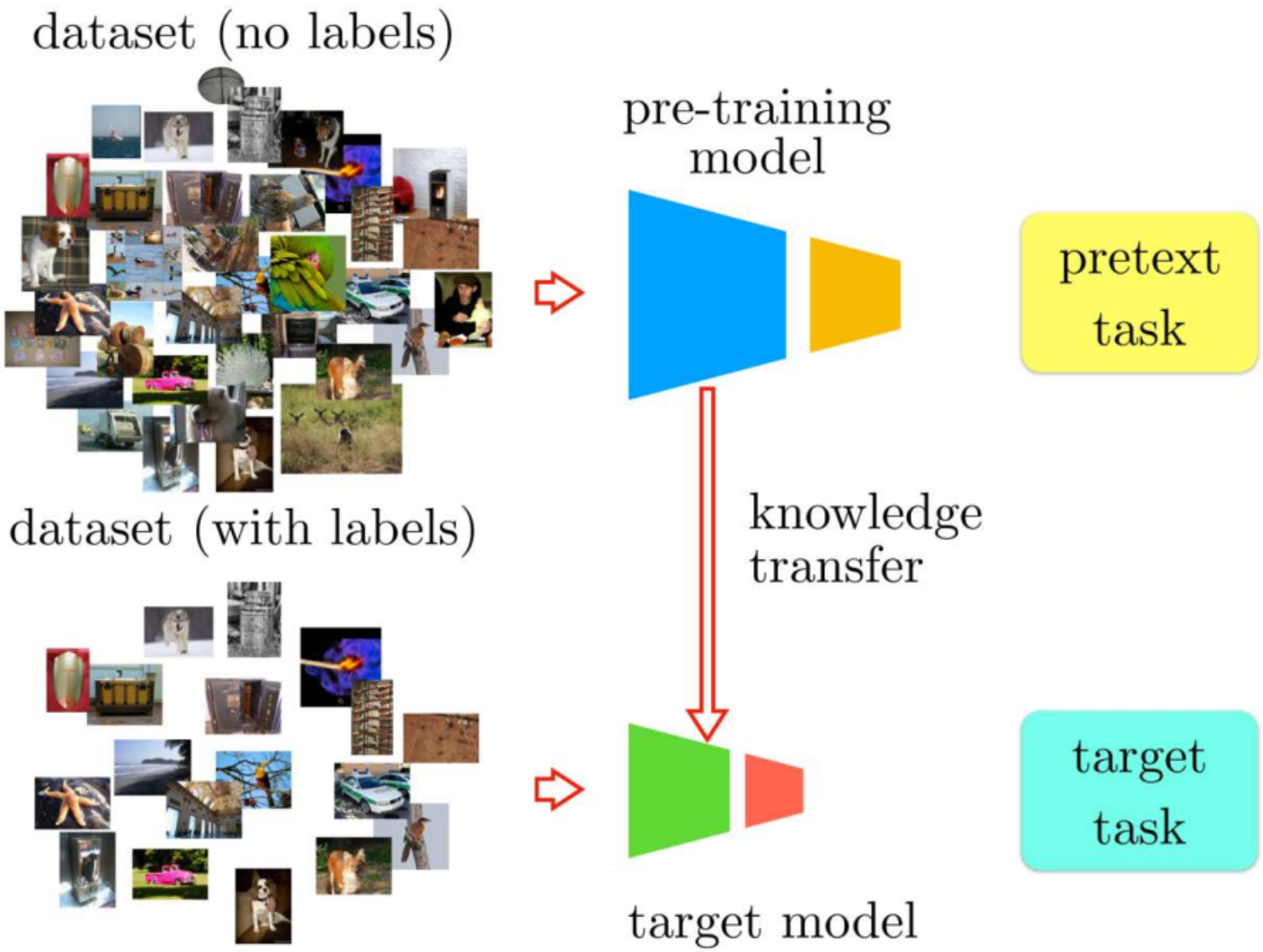

1. train the network initially on a very large set of "somewhat" related images
2. fix the lower levels of the network
3. for a new, different, and smaller set of specialised data, tune the upper network to better fit the new data

# Pre-Training: Fix Features, cont.




1. train the network initially on a very large set of "somewhat" related images
2. fix the lower levels of the network
3. for a new, different, and smaller set of specialised data, tune the upper network to better fit the new data

# Pre-Training Method



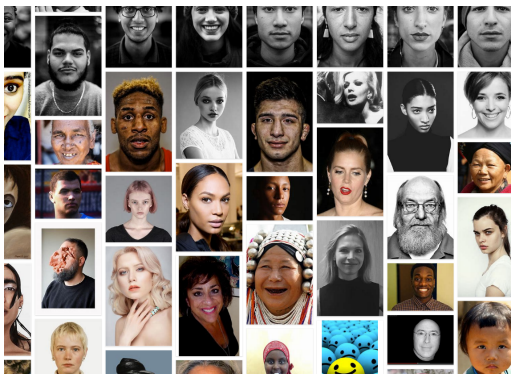

from Noroozi, Vinjimoor, Favaro, Pirsiavash, CVPR 2018

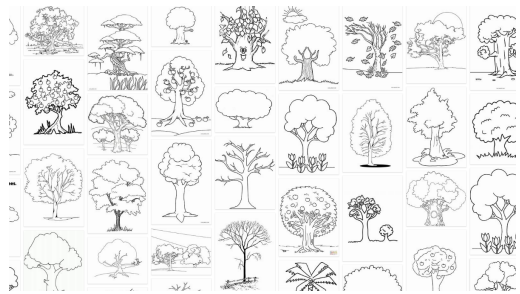
# Pre-Training: Discussion

- pre-training is a way of **initialising** and/or a way of **providing lower-level features**
  - able to make use of very large unlabelled datasets for text and images
- we pre-train on a large set of data once, and reuse the pre-trained model many times
- for lower-level features, you would hope the tasks have "lower-level" similarity
- if you have less training data for your new task, leave more of the lower-level layers intact
  - reduce the capacity of the network to be tuned, since there is less data available to do it

# Pre-Training: Discussion, cont.

How useful are these three images sets for subsequently **learning flowering plants?**





# Self-supervision

- to do pre-training, we need a task for learning
- embedding methods are an early precursor
- pre-training should build lower-level features useful for subsequent target classes

Self-supervision: an artificial task created for the purposes of learning a network useful as a pre-trained network.

- called "self" supervised since the task is created automatically

# Self-supervision, cont.

Self-supervision: an artificial task created for the purposes of learning a network useful as a pre-trained network.

- examples for image recognition:
  - color a B/W image
  - fill-in missing patches ("image in-painting")
  - object classification from very broad image class
- examples for text classification:
  - predict missing words
  - forwards and backwards order

# Self-supervision, cont.

Self-supervision: an artificial task created for the purposes of learning a network useful as a pre-trained network.

- examples for image recognition:
  - color a B/W image
  - fill-in missing patches ("image in-painting")
  - object classification from very broad image class
- examples for text classification:
  - predict missing words
  - forwards and backwards order
- generally, the self-supervision task should be (1) richer and more refined than or (2) similar to subsequent target tasks

# Self-supervision, cont.

Self-supervision: an artificial task created for the purposes of learning a network useful as a pre-trained network.

- examples for image recognition:
  - color a B/W image
  - fill-in missing patches ("image in-painting")
  - object classification from very broad image class
- examples for text classification:
  - predict missing words
  - forwards and backwards order
- generally, the self-supervision task should be (1) richer and more refined than or (2) similar to subsequent target tasks
- fundamentally different to component models builts as a latent variable models (e.g., matrix factorisation)

# Pseudo-likelihood and Self-supervision

A pseudo-likelihood (Besag 1975) is an approximation to the joint probability distribution of a collection of random variables using univariate conditionals:

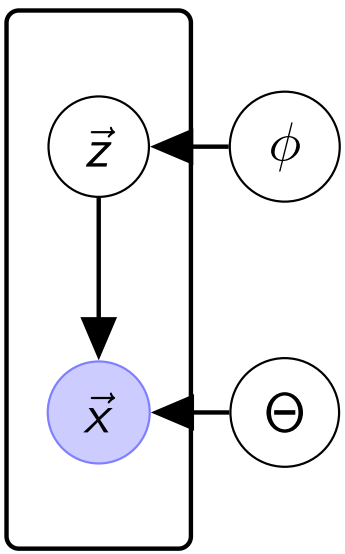
$$\tilde{p}(X|\Theta) \equiv \prod_{x \in X} p(x \mid X \setminus \{x\}, \Theta)$$

- it is easily computed because the individual conditionals can usually be easily marginalised.
  - used to train Markov networks in neural networks
- Maximising pseudo-likelihood is known to be consistent with maximising likelihood in the limit of infinite data.
- Pseudo-likelihood can be viewed as a simplified theoretical view of self-supervision.
- But an alternative view is of representation learning.

# Artificial Faces

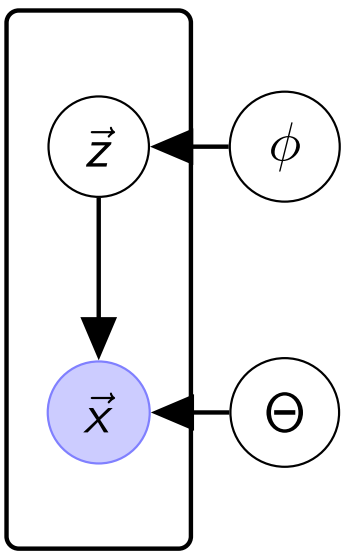
Early Generative Adversarial Networks (GANs) were trained to generate new faces given 1000s of examples.

# Generative Deep Learning



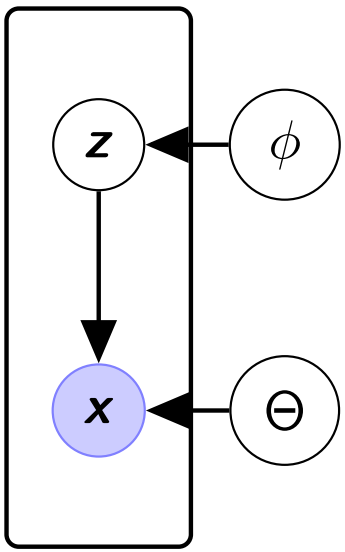

- the Generative Adversarial Networks (GAN) and Variational Autoencoder (VAE) implement the same model but with different algorithms
  - model is on the left
  - GAN doesn't explicitly find $\vec{z}$ during training

# Generative Deep Learning




- the Generative Adversarial Networks (GAN) and Variational Autoencoder (VAE) implement the same model but with different algorithms
  - model is on the left
  - GAN doesn't explicitly find $\vec{z}$ during training
- this is a generalisation of LDF, PMF, ICA, PCA, etc., all the matrix factorization models
- in our experience, individual elements of the latent vector $\vec{z}$ are encouraged to be independent
  - more independence means higher capacity model

# Variational Autoencoder, cont.

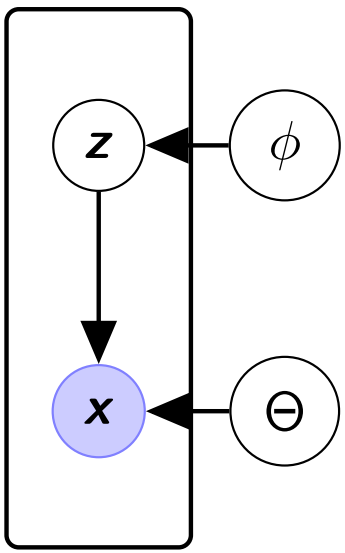



- The original VAE algorithm is similar to the standard variational method
  - $q(Z|X)$ is fit using Equation (2), not (1)
  - so in principle it can optimise
  - surprisingly good scaling properties using amortisation
- very good topic model variants exist
  - needed Bayesian guys to do a hybrid Deep-Bayesian method

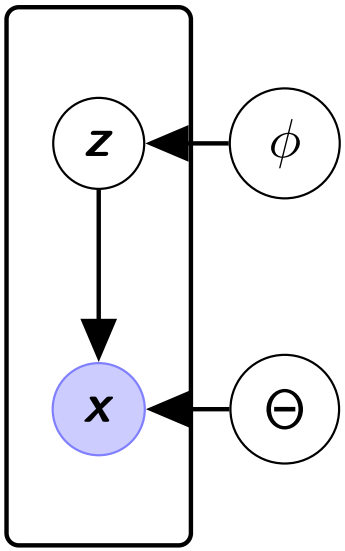
# Generative Deep Learning, cont.

- the original GAN is (almost) equivalent to minimising conjugate formulation of J-S divergence between data and model
  - Nowozin's GAN tutorial at MLSS 2018
  - data distribution in this case is

$$p(x|data) = \frac{1}{I}\sum_{i=1}^{I} \delta_{x_i=x}$$



# Generative Deep Learning, cont.

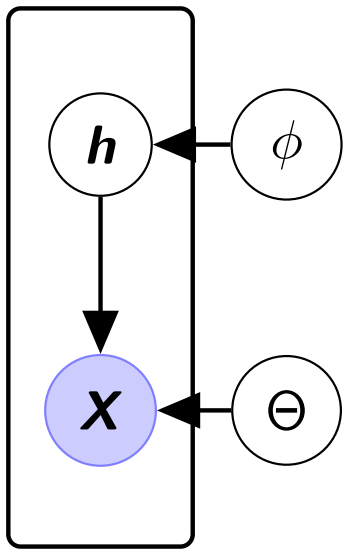


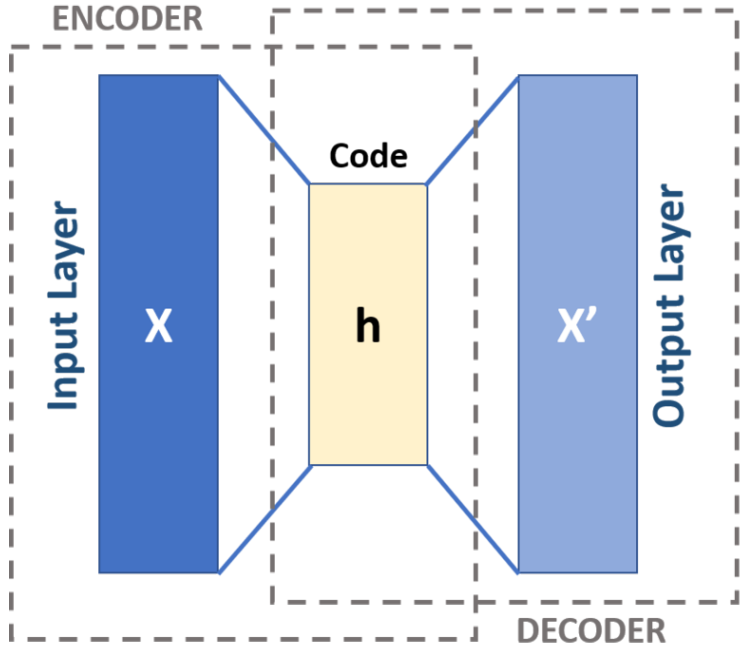

- the original GAN is (almost) equivalent to minimising conjugate formulation of J-S divergence between data and model
  - Nowozin's GAN tutorial at MLSS 2018
  - data distribution in this case is

$$p(x|data) = \frac{1}{I}\sum_{i=1}^{I}\delta_{x_i=x}$$

  - is the adversarial framework a mathematical digression?
  - applying K-L divergence to same data distribution is equivalent to maximum likelihood!
- GANs are judged completely differently to other learned models
- state-of-the-art GANs use transport metrics and ensembles (I'm told)

# Encoder-Decoder Versus VAE




right figure from Wikipedia

The encoder-decoder framework can be interpreted probabilistically as a GAN/VAE type model where the latent variable $\vec{h}$ is discrete instead of real-valued.

# Disentangling Concepts

- disentangling is a term popularised in Bengio, Courville, and Vincent's classic 2012, IEEE Trans PAMI paper
- term is somewhat vaguely defined in the literature!
- we want discovered latent variables of a GAN/VAE to be "independent causally" w.r.t. the task

e.g. consider the task of digit recognition

- color, rotation, shift, thickness, font style are independent dimensions we would like to discover that are irrelevant to the digit classification

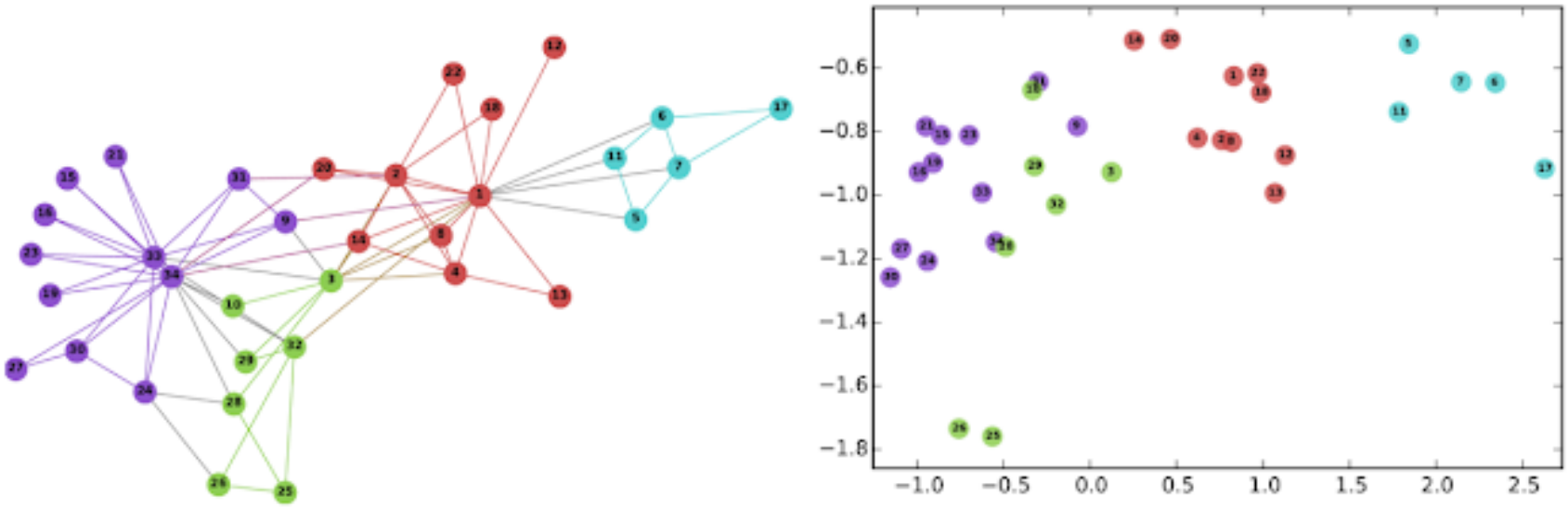
# Disentangling Concepts

- disentangling is a term popularised in Bengio, Courville, and Vincent's classic 2012, IEEE Trans PAMI paper
- term is somewhat vaguely defined in the literature!
- we want discovered latent variables of a GAN/VAE to be "independent causally" w.r.t. the task

e.g. consider the task of digit recognition

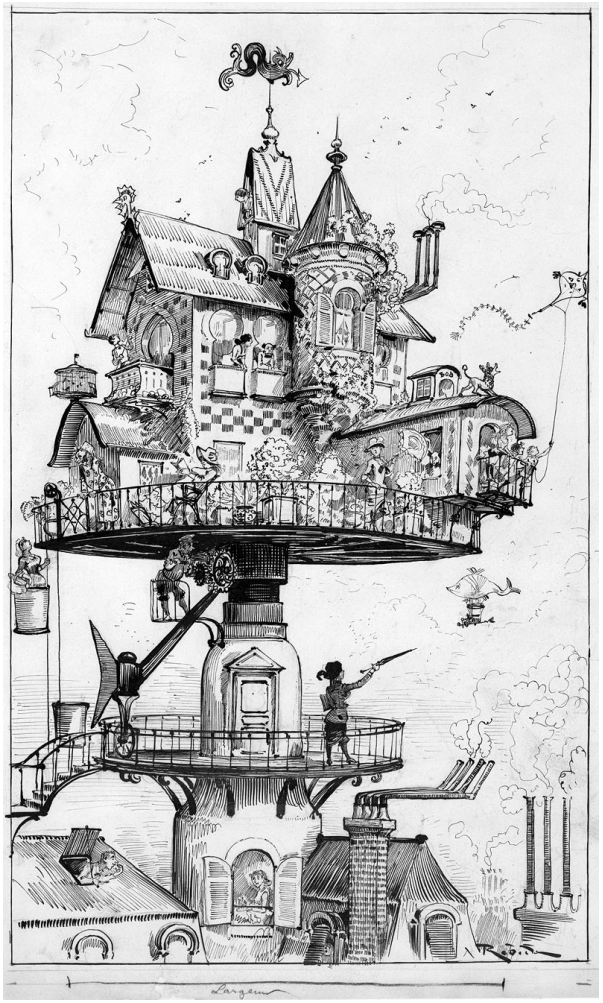
- color, rotation, shift, thickness, font style are independent dimensions we would like to discover that are irrelevant to the digit classification $\longrightarrow$ disentangle them!

# Graph Embeddings

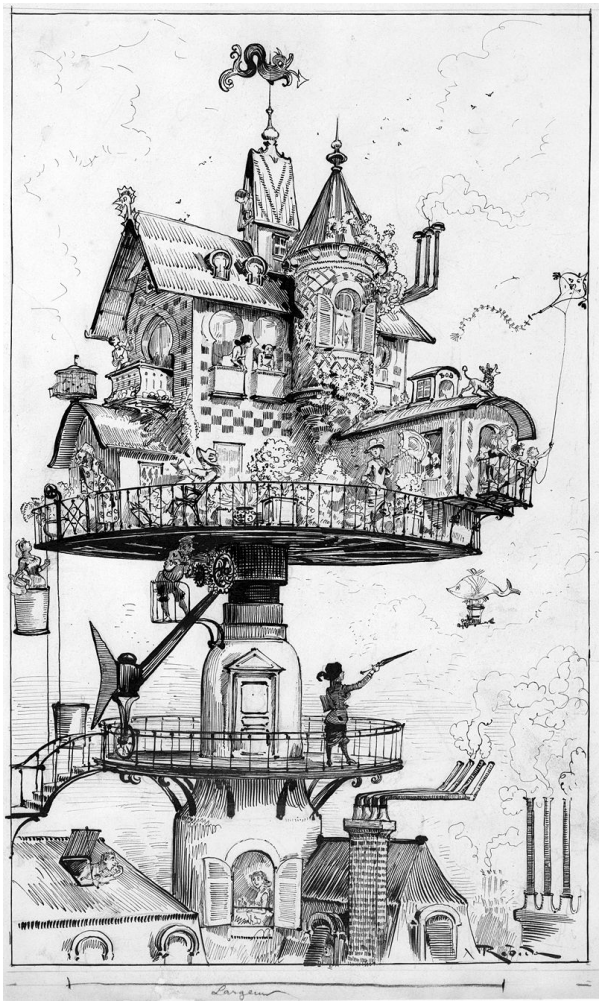



- techniques similar to word embeddings have been developed for graphs
- similar techniques apply: cost functions, negative sampling,
- nodes in graphs also have extensive side information
  - "hospital patient" node in a graph may have their socio-economic data attached as an attribute of the node

# Outline



# Outline



# Consider Object Recognition

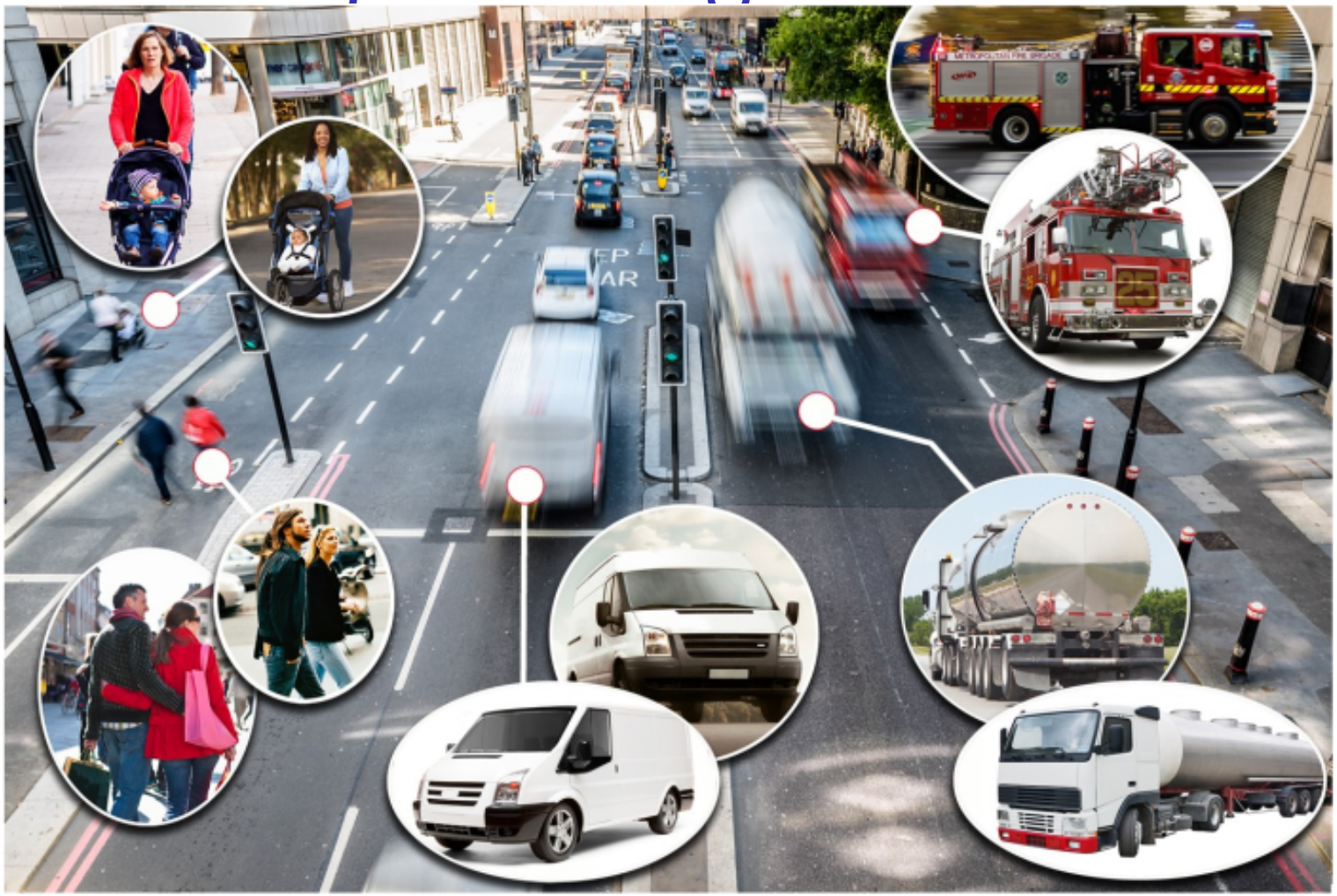
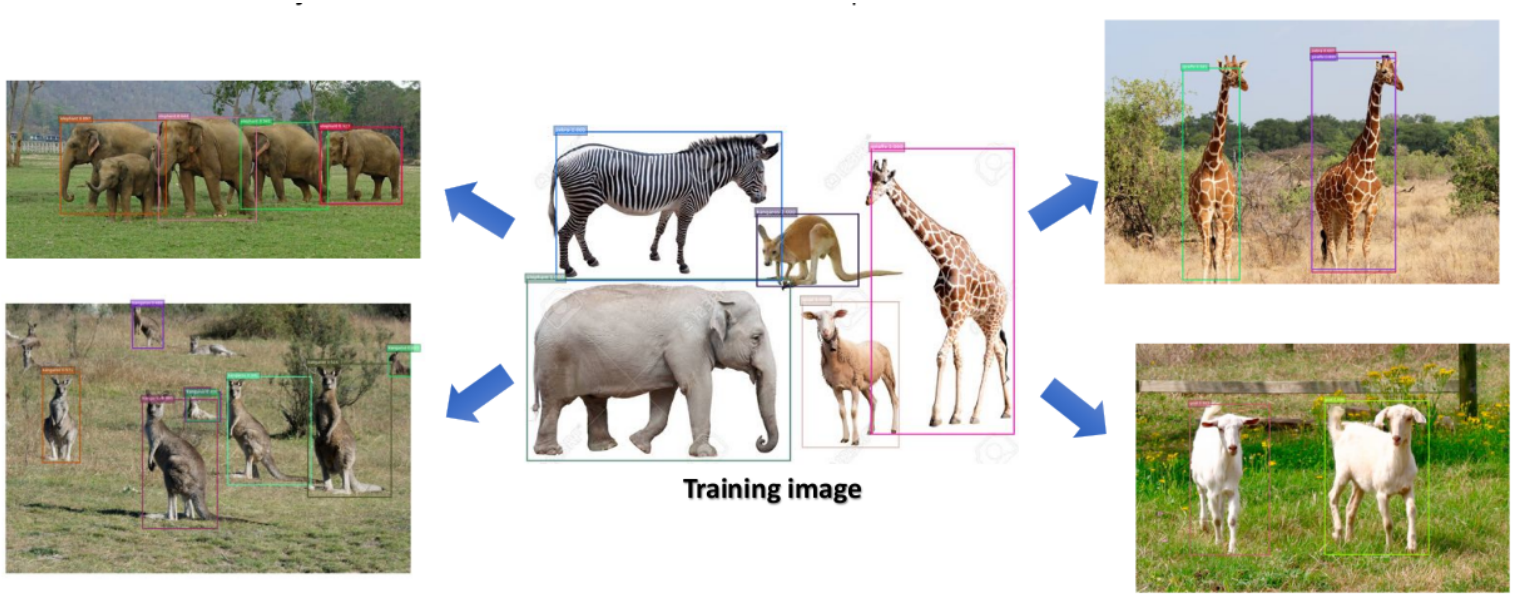


image from Heiko Hoffmann, HRL Laboratories, DARPA 2019

Wouldn't it be good if we could learn all the different object classes with just a few examples!

# One-Shot Learning




from

Karlinsky et al, CVPR 2019

To do large scale object recognition and scene understanding in images, it would be good to have one-shot (or "few-shot") learning working.

# One-Shot Learning, cont.

**zero-shot:** recognise given a description rather than an example, i.e., this is IR

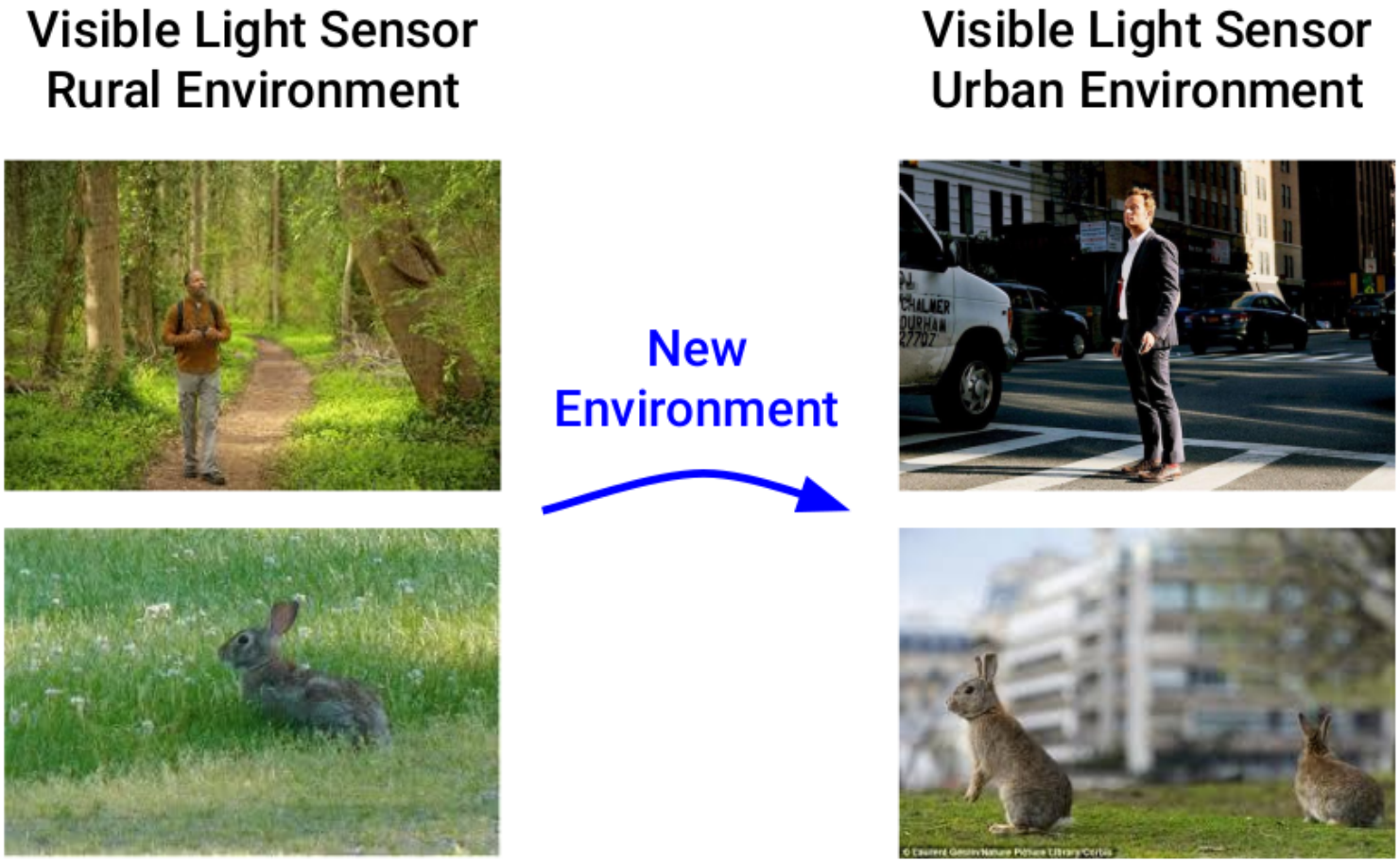
# One-Shot Learning, cont.

**zero-shot:** recognise given a description rather than an example,
i.e., this is IR

**few-shot:** recognise from a few examples,
i.e., this is data poor learning

- one/few-shot have a wide variety of approaches
- research largely in image recognition
- current focus is the continuum between 0-1-few-many
- requires a trade-off of prior knowledge and data

# Covariate Shift




figure

from Eisenberg, Embedded Intelligence, DARPA 2019

Changing the context or background distributions, makes recognition different.

Called covariate shift, and a form of domain adaptation.

# Covariate Shift, cont.

**Problem setup:** target variables or input data changes in distribution; for prediction of $y_i$ from features $\boldsymbol{x}_i$

- in covariate shift, $p(\boldsymbol{x}_i)$ changes, not $p(y_i|\boldsymbol{x}_i)$
- however, new data after shift will occur outside the "comfort zone" of the initial trained system
- if training builds a causal model, it should be robust to covariate shift
  - but rarely done!

# Task Types

(my take on the old ML version)

**Shift:** target variables or input data changes in distribution; for prediction of $y_i$ from features $\boldsymbol{x}_i$:

**Online:** learning and evaluation is done incrementally

**Multi-task:** several/many related tasks are done

**Life-long:** new, somewhat related tasks are constantly appearing

**Human in the Loop:** during learning, varied forms of human involvement:

- can choose to label via active learning
- similar to crowd-sourcing (i.e., a flawed oracle)

**Recognition:** identifying an object within a broader context

- objects in vision, named-entities in text

# Task Types

(my take on the old ML version)

**Shift:** target variables or input data changes in distribution; for prediction of $y_i$ from features $\boldsymbol{x}_i$:

**Online:** learning and evaluation is done incrementally

**Multi-task:** several/many related tasks are done

**Life-long:** new, somewhat related tasks are constantly appearing

**Human in the Loop:** during learning, varied forms of human involvement:

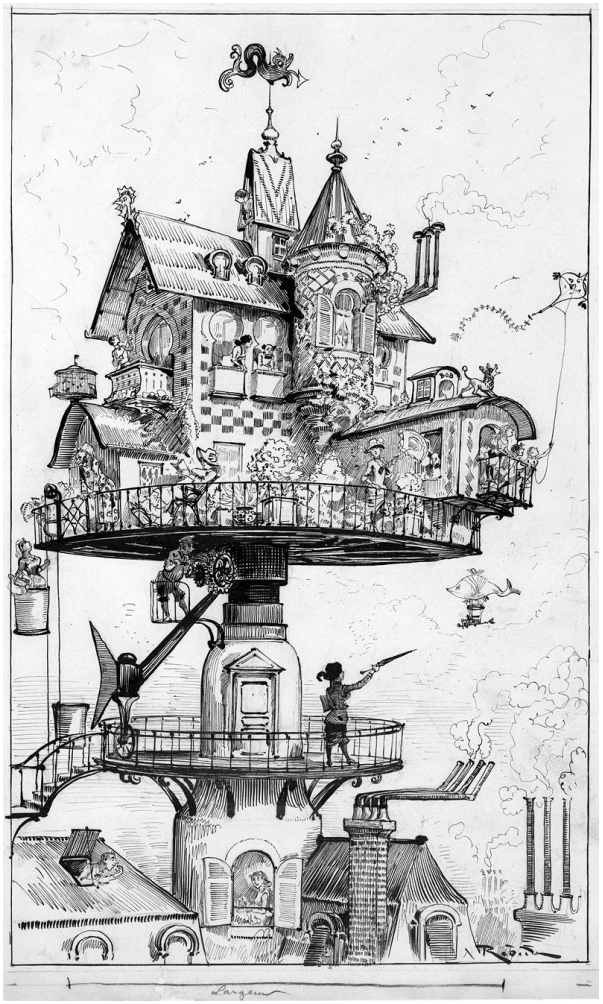
- can choose to label via active learning
- similar to crowd-sourcing (i.e., a flawed oracle)

**Recognition:** identifying an object within a broader context

- objects in vision, named-entities in text

learning to learn is usually situated in these richer learning tasks

# Outline



# Weak Supervision

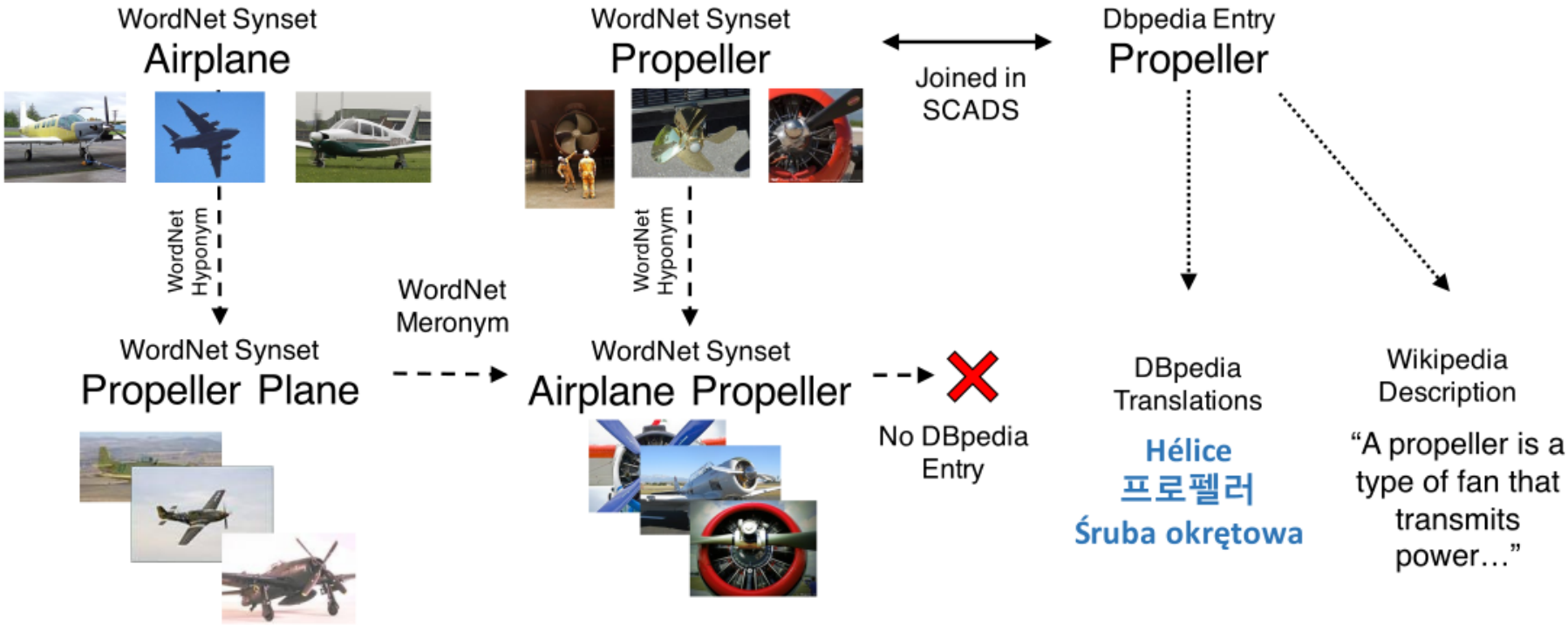



figure from Stephen Bach, Brown University, DARPA 2019

Different kinds of "weak" information about a concept can be used to support learning $\longrightarrow$ weak supervision.

# Semi-Supervised Learning

# Entropy Minimisation for Semi-Supervised Learning

# Supervision Types

(my take on the old ML version)

**Supervised:** classes or target variables for prediction are supplied with the data

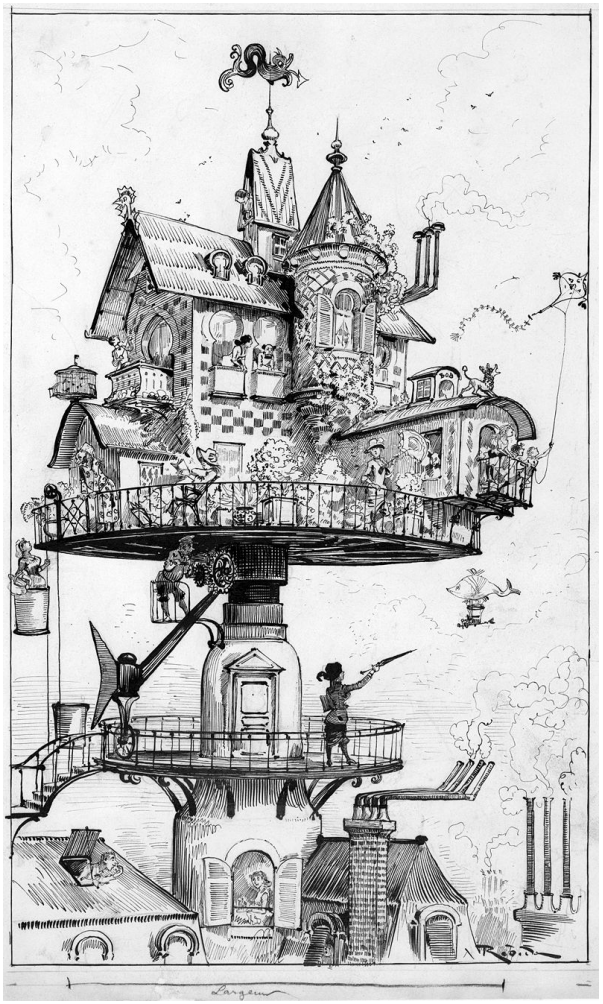
**Unsupervised:** no supervision is given at all; learn about the data generally

**Semi-supervised:** classes or target variables for prediction are supplied with the data for only a (smaller) subset of data

**Weakly supervised:** incomplete or approximate supervision supplied with the data

**Self-supervised:** an artificial task is created for the purposes of pre-training

# Outline



# Background Knowledge Types

Related tasks: useful for selection of data for pre-training and related systems for domain adaptation, ...

Features: causal relationships, invariances to support augmentation, as trained by pre-training, ...

Data augmentation: invariants, monotonicities other properties of data ...

Weak supervision: special case of related tasks, useful for partial, initial or filtering models, ...

Task descriptions: used for weak-supervision, selection of related tasks, ...

Regularisation: entropy minimisation, consistency regularisation, prior rate alignment, ...

# Background Knowledge Types

Related tasks: useful for selection of data for pre-training and related systems for domain adaptation, ...

Features: causal relationships, invariances to support augmentation, as trained by pre-training, ...

Data augmentation: invariants, monotonicities other properties of data ...

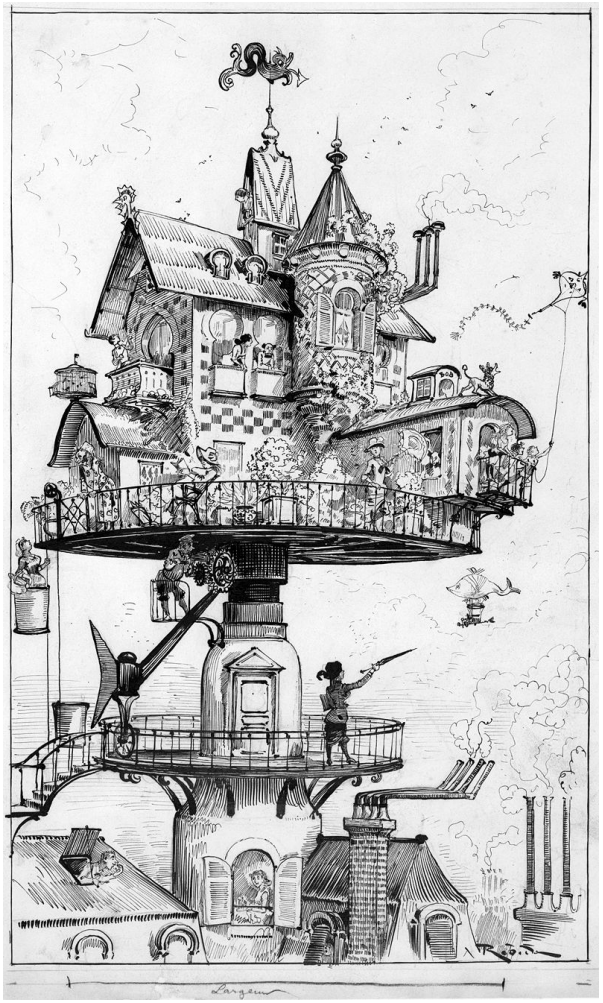
Weak supervision: special case of related tasks, useful for partial, initial or filtering models, ...

Task descriptions: used for weak-supervision, selection of related tasks, ...

Regularisation: entropy minimisation, consistency regularisation, prior rate alignment, ...

These often work orthogonally to tasks

# Outline



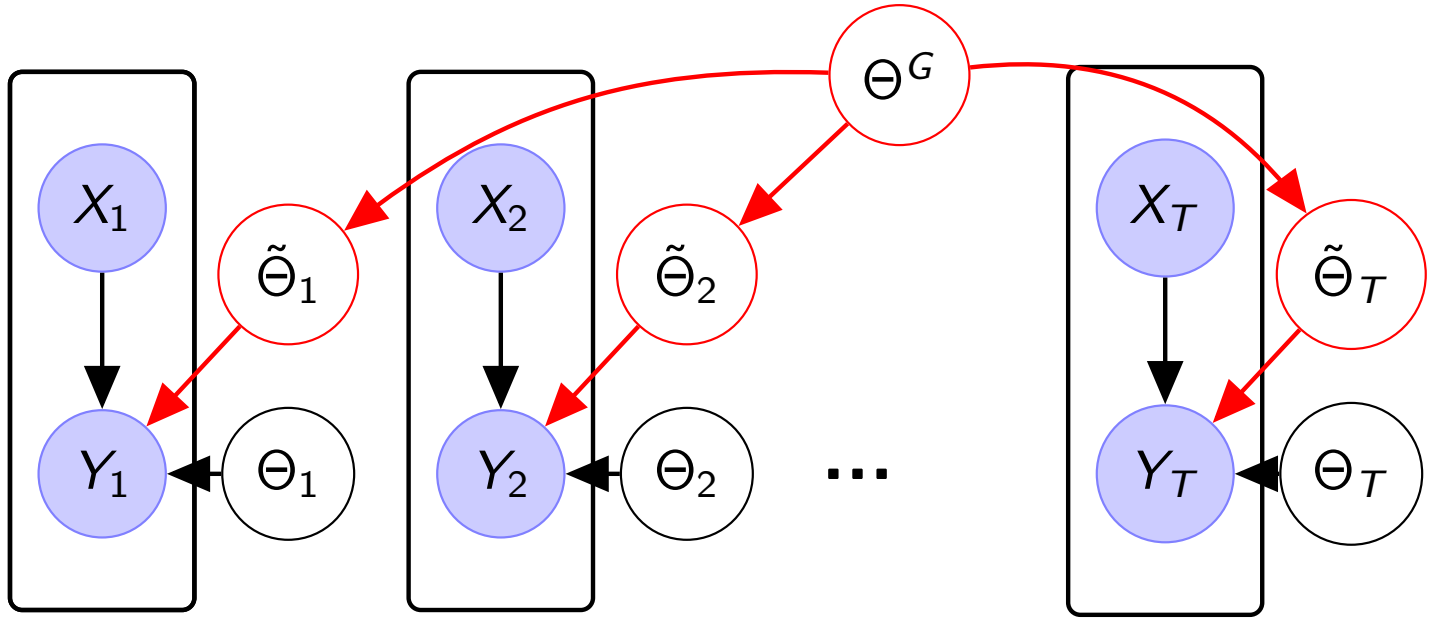
# Meta-Learning

**Meta-Learning:** where learning is applied to support the task of learning itself.

- meta learning strongly inspired by the human experience
  - my motivation for entering Machine Learning in 1984
- is patchwork of different techniques, not a single consistent method
  - reinforcement learning
  - hierarchical Bayes
  - architectural tricks in deep learning
- large groups are building pipelines and architecture
  - one of a few key "hot" areas in ML

# Multi-Task Learning (MTL)

We can rebrand hierarchical Bayesian modelling as multi-task learning.

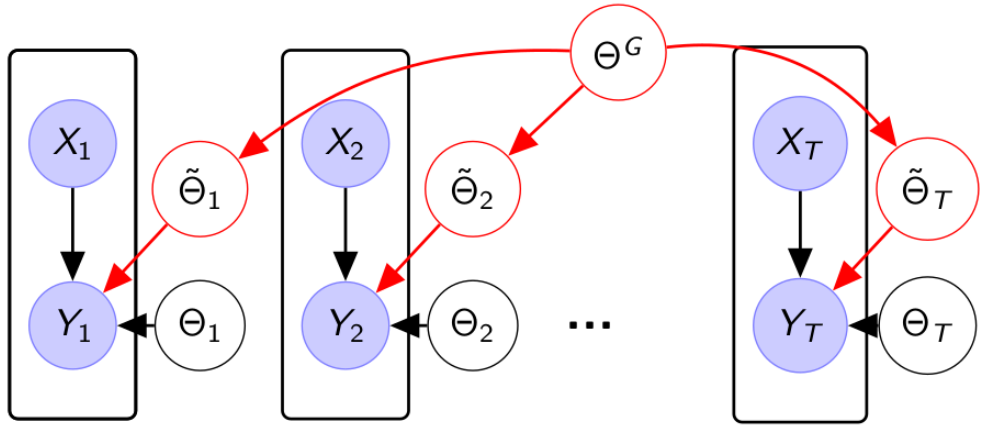



$\Theta_1 - \Theta_T$: task specific parameters

$\Theta^G$: shared parameters

$\tilde{\Theta}_1 - \tilde{\Theta}_T$: task specific instantiation of shared parameters

# Simple Meta-Learning




Shared parameters $\Theta^G$ could be:

- hyper-parameters of optimiser (e.g., step size)
- parameter initialisations
- lower-level part of a deep NN

Each of these has a huge recent literature!

# Model Agnostic Meta-Learning

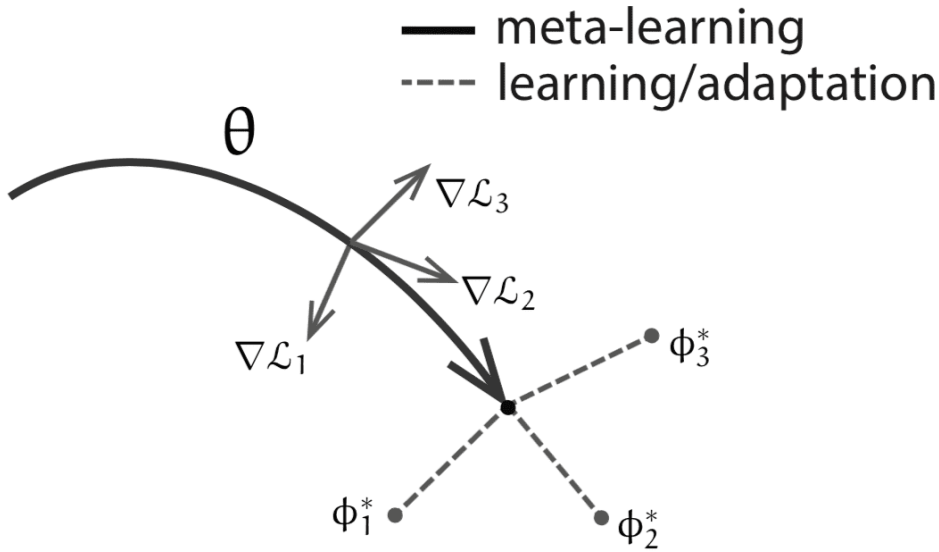
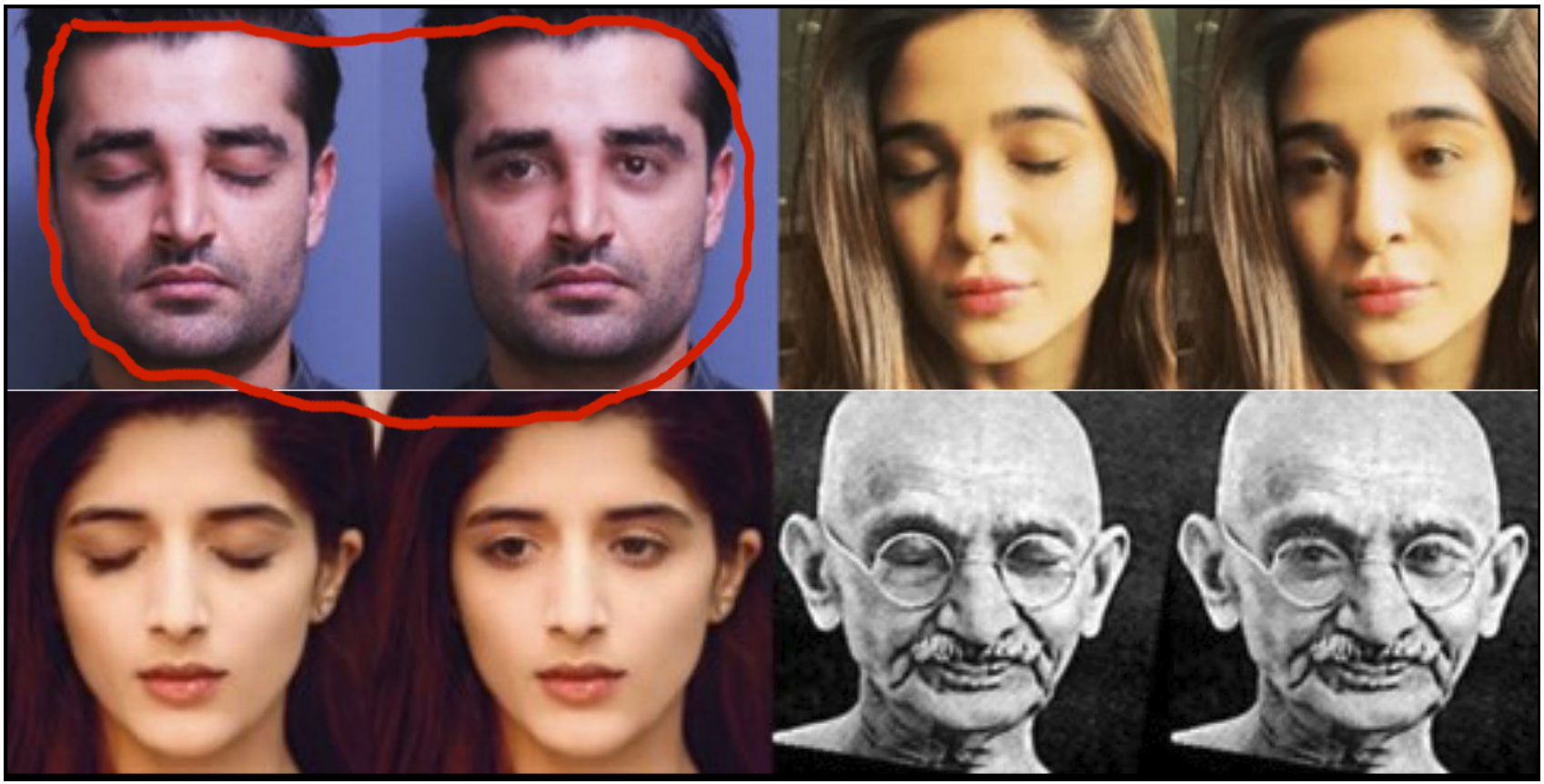


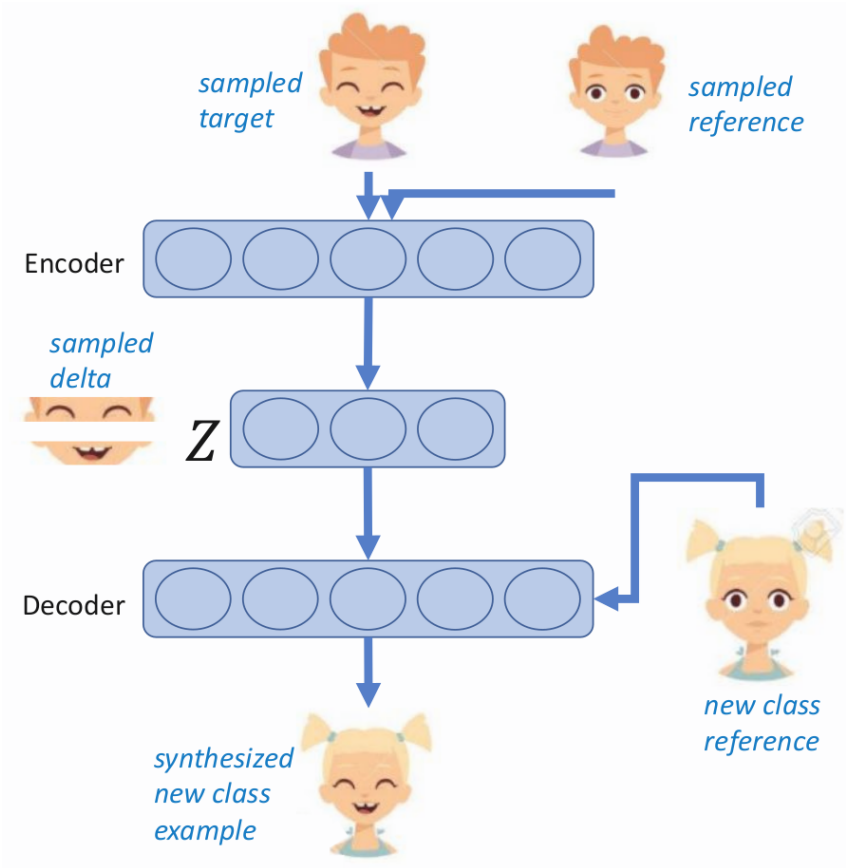

Figure 1: Illustrative diagram of our model-agnostic meta-learning algorithm (MAML), which optimizes for a representation θ that can quickly adapt to new tasks.

- learn parameter initialisations (using one-shot framework)
- see Finn, Abbeel and Levine, 2017, arXiv:1703.03400

# Altering Faces: Style Learning



# Learning to Augment Data




from Schwartz et al, NeurIPS 2018, Delta-encoder

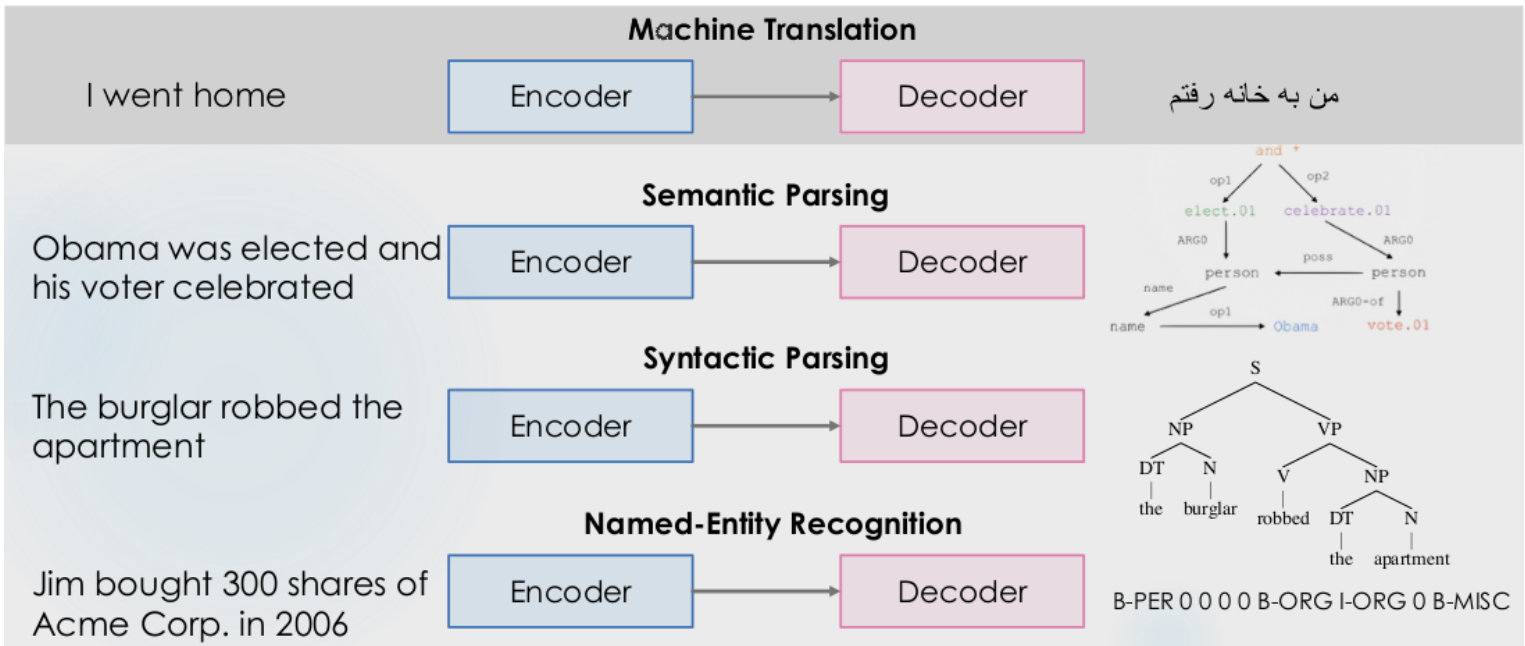
- learn the variance, spread or difference of typical examples from related domains
- apply the variance/spread/difference to the new data
- often uses GANs for the learning
- can be used as a hybrid of weak supervision
  e.g., adding estimated bounding boxes to objects in images

# Neural Machine Translation (NMT)

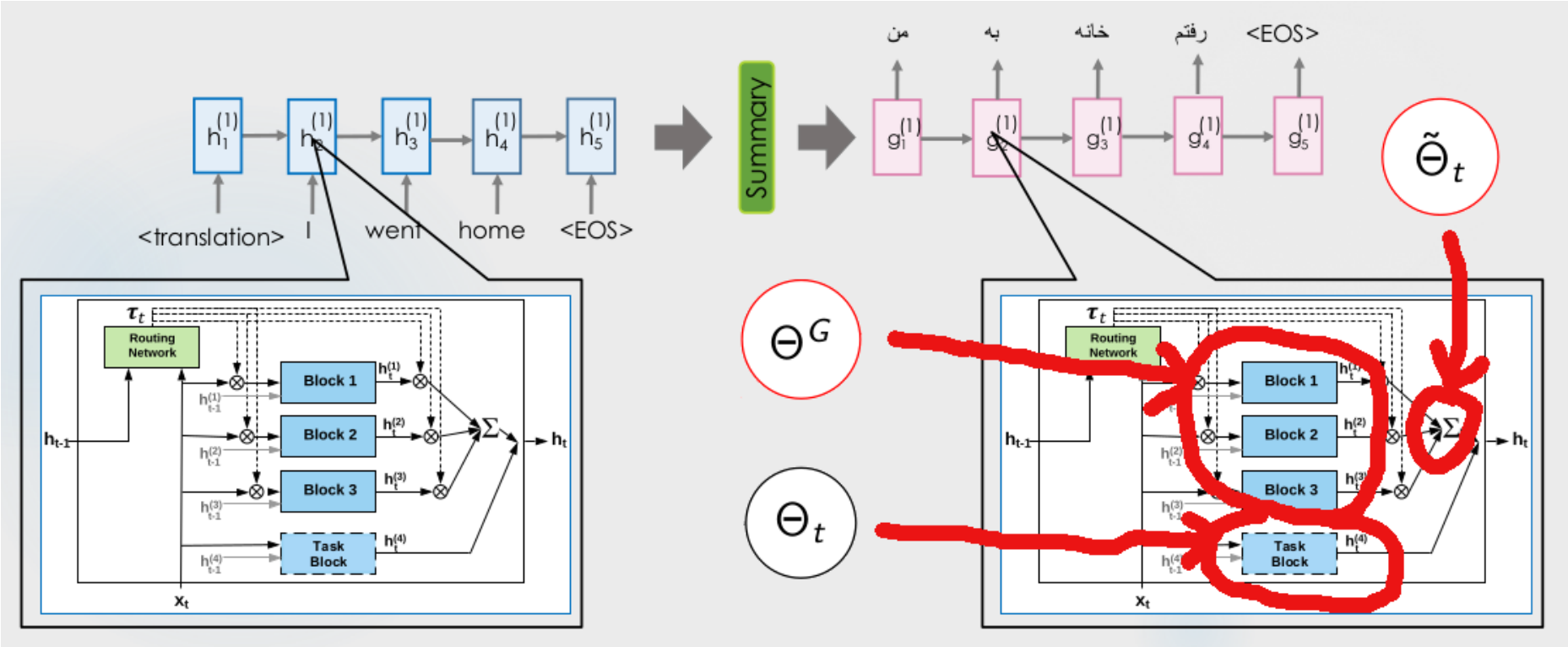
ZareMoodi, Buntine, Haffari *ACL* 2018

- Bilingually low-resource scenario: large amounts of bilingual training data is not available.

IDEA: Use existing resources from other tasks and train one model for all tasks using multi-task learning (MTL).



# NMT: Multi-Task Model




- Block-1 to Block-3 are task independent components, $\Theta^G$ the shared common knowledge for MTL
- Routing-Network controls their use on a task to create $\tilde{\Theta}_t$
- task specific parameter is $\Theta_t$

# Questions?

Figure source: http://milewalk.com/mwblog/4-worst-job-interview-questions/