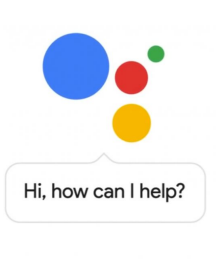
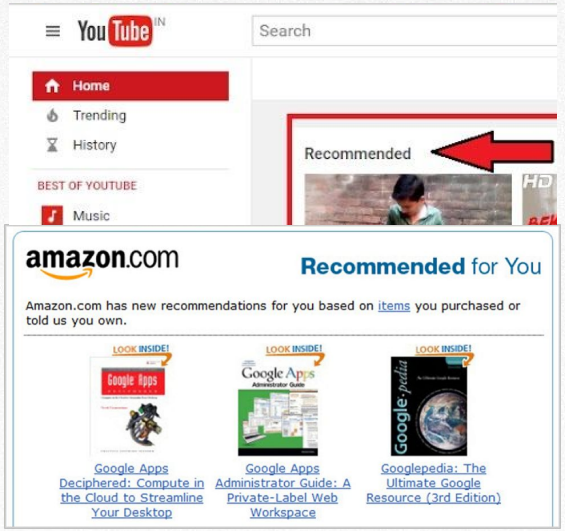
Interpretable Models in Probabilistic Deep Learning

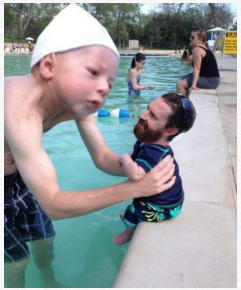
Hyunjik Kim

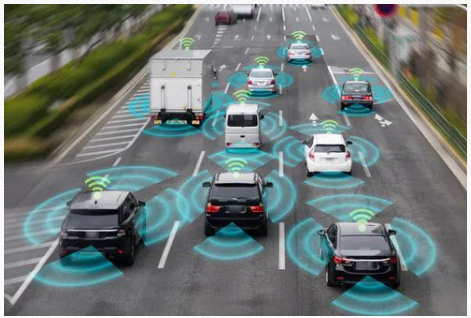
University of Oxford

Deep Learning in the Wild









DeepMind

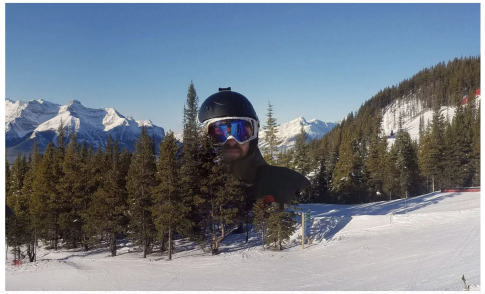
Failure modes

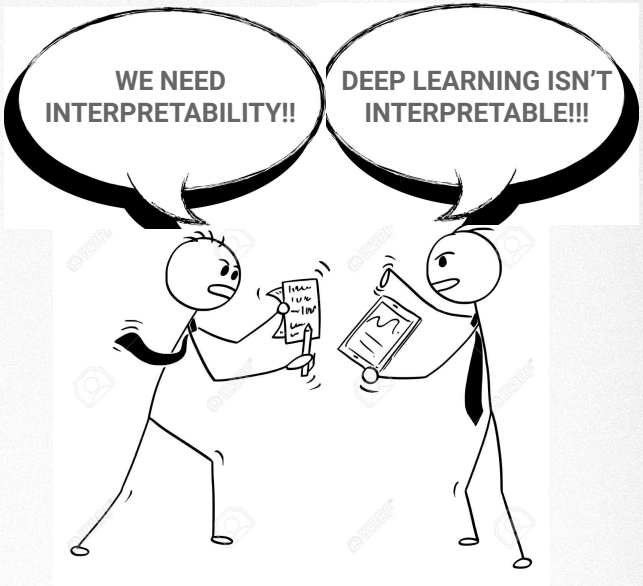
Amazon scraps secret AI recruiting tool that showed bias against women

Google Photos' AI Panorama Failed in the Best Way

JAN 23, 2018 MICHAEL ZHANG

Share 3.2K Tweet 27 COMMENTS





What is Interpretability?

Definition proposed by [1]:

“ability to explain or to present in understandable terms to a human”

But…

- What do we mean by “understandable terms”?
- How do we draw the line between “explained” and “unexplained”?

Maybe more helpful to look at:

- **What do we want** from interpretability?
- **What are the properties** can be called interpretable?

[1] Doshi-Velez & Kim - Towards a Rigorous Science of Interpretable Machine Learning

What do we want from Interpretability? [2]

- **Trust**: confidence of model performance in real scenarios
- **Transferability**: robustness to adversarial examples & ability to transfer learned skills to unfamiliar situations
- **Fair and ethical decision making**: decisions conform to ethical standards
- **Causality**: help form causal hypotheses that can be tested
- **Informativeness**: provide useful information for human decision-making

[2] Lipton - The Mythos of Model Interpretability

Properties of Interpretability [2]

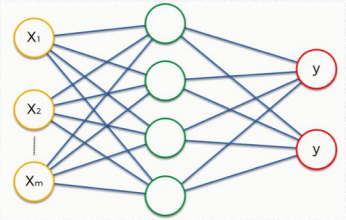
- **Transparency**: (humans) can understand the mechanism of model/algorithm
  - **simulatability**: easily understandable computation
  - **decomposability**: each part of model (e.g. parameters, features) admits an intuitive explanation
  - **algorithmic transparency**: easy to determine whether the model will or will not work on unseen data points / datasets
- **Post-hoc interpretability**: (humans) can obtain useful information for an end task
  - **text explanation**
  - **visualisation:** qualitative understanding of model
  - **local (per-data point) explanation**
  - **explanation by example** e.g. finding points which the model views to be similar

[2] Lipton - The Mythos of Model Interpretability

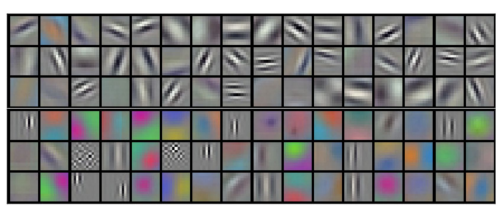
Interpretability of (Probabilistic) Deep Learning

Transparency: (humans) can understand the mechanism of model/algorithm

- **simulatability**: easily understandable computation



- **decomposability**: each part of model (e.g. parameters, features) admits an intuitive explanation

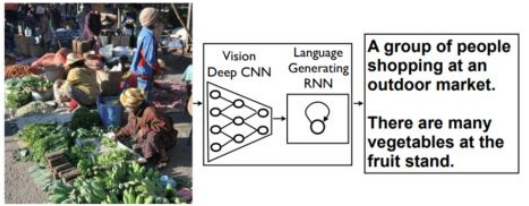


- **algorithmic transparency**: easy to determine whether the model will or will not work on unseen data points / datasets

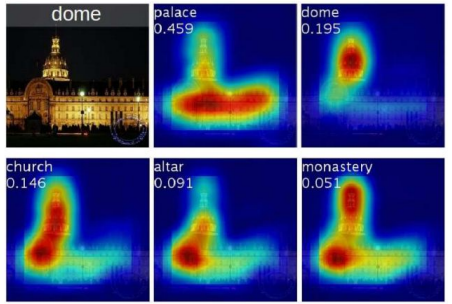
Interpretability of (Probabilistic) Deep Learning

Post-hoc interpretability: (humans) can obtain useful information about model's mechanism and/or its predictions

- **text explanation**



- **visualisation**: qualitative understanding of model

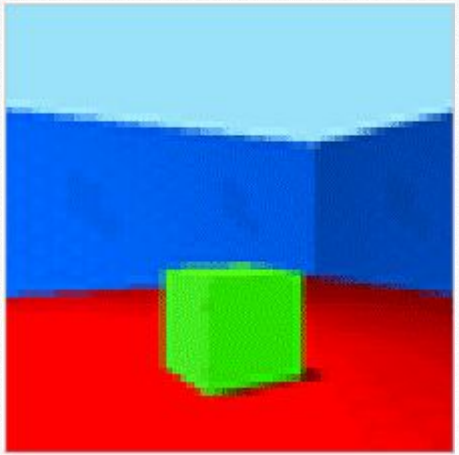


- **local (per-data point) explanation**
- **explanation by example** e.g. finding points which the model views to be similar

Examples

(Non-representatitve) examples of DL models with interpretable properties:

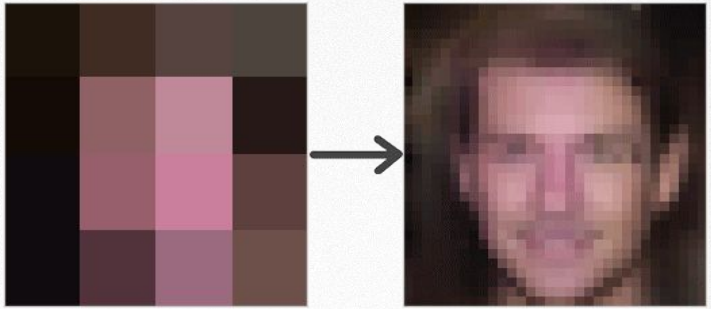
FactorVAE



Attentive Neural Process

Bottom half prediction

Super-resolution



Disentangling by Factorising

ICML '18

Hyunjik Kim, Andriy Mnih

What is disentanglement??

- There is NO canonical definition that everyone agrees on.
- We **assume data generated from independent factors of variation** and crudely define a ‘disentangled representation’ to mean:

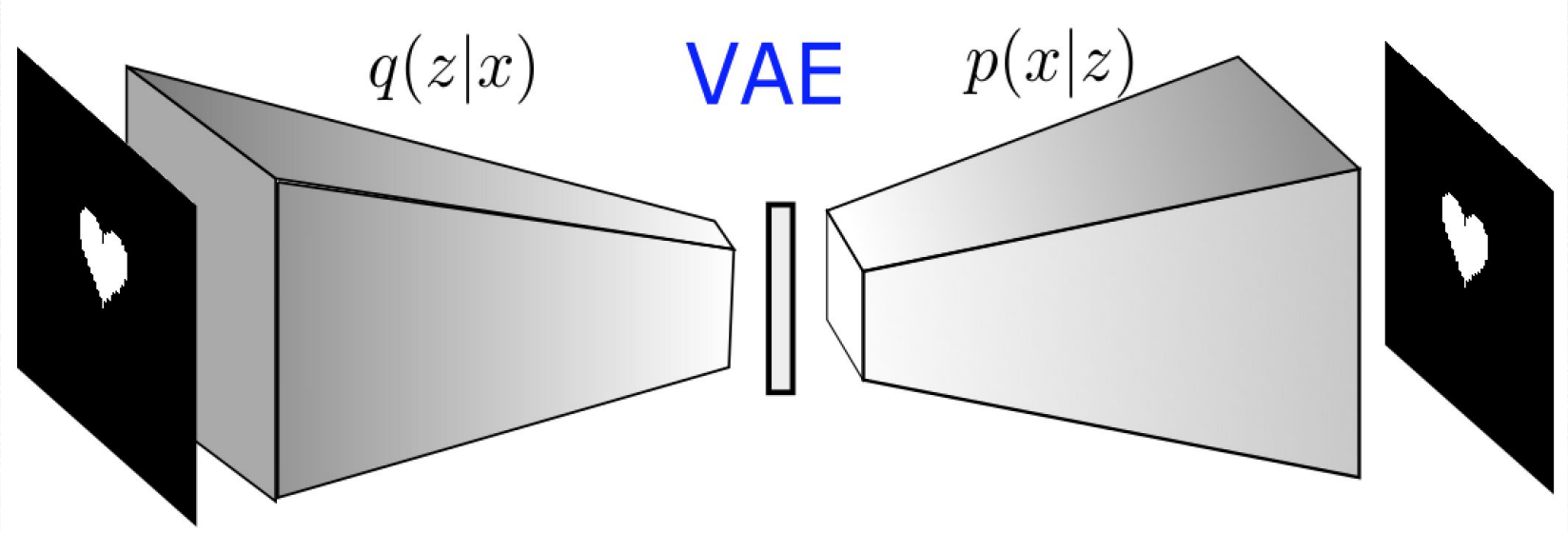
  “Change in one dimension corresponds to change in one factor of variation in the data”

- For simplicity, we ignore the possibility of:
  - correlations among the factors of variation
  - hierarchy in the factors of variation
  - many to one mapping between a combination of factors and a data point (over-representation)

It’s already hard enough!

Variational Autoencoder (VAE)

Summary



Marginal Posterior of VAEs

- Code distribution: $q(z) = \int p_{data}(x) q(z|x) dx = \frac{1}{n} \sum_{i=1}^{n} q(z|x^{(i)})$
- Sample via ancestral sampling: $z \sim q \Leftrightarrow x \sim p_{data}, z \sim q(\cdot|x)$

Marginal Posterior of VAEs

- Why is this relevant for disentangling?
  - Typically want code distribution to be independent in the dimensions.
  - i.e. want :

$$q(z) = \prod_{j=1}^{D} q(z_j)$$

Beta-VAE

Optimisation Objective

Minimise $\forall x^{(i)} \sim p_{data}$ [3] :

$$\underbrace{-\mathbb{E}_{q(z|x^{(i)})}[\log p(x^{(i)}|z)]}_{\text{Reconstruction Error}} + \underbrace{\beta KL(q(z|x^{(i)})||p(z))}_{\text{Complexity Penalty}}$$

[3] I. Higgins et al - beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework

KL decomposition

The KL term decomposes as follows [4] :

$$KL(q(z|x)||p(z)) = I(x;z) + KL(q(z)||p(z))$$

$KL(q(z)||p(z))$ - Pushes $q(z)$ towards an independent distribution, thereby encouraging independence in the dimensions.

$$I(x;z) = \mathbb{E}_{p_{data}(x)}[KL(q(z|x)||q(z))]$$

- Acts as **information bottleneck**
- Forcing efficient use of codes and hence disentangling?
- Too heavy penalty (high β) leads to poor reconstruction (**Tradeoff**)

[4] M. Hoffman et al - Elbo surgery: yet another way to carve up the variational evidence lower bound

Motivation

Can we get a better tradeoff between good reconstruction and disentangling?

Idea: Keep VAE loss, and directly penalise $KL(q(z)||\prod_j q(z_j))$

New loss for **FactorVAE**:

$$\underbrace{-\mathbb{E}_{q(z|x^{(i)})}[\log p(x^{(i)}|z)] + KL(q(z|x^{(i)})||p(z))}_{\text{VAE loss}} + \underbrace{\gamma KL(q(z)||\prod_{j=1}^{D} q(z_j))}_{\text{Total Correlation(TC)}}$$

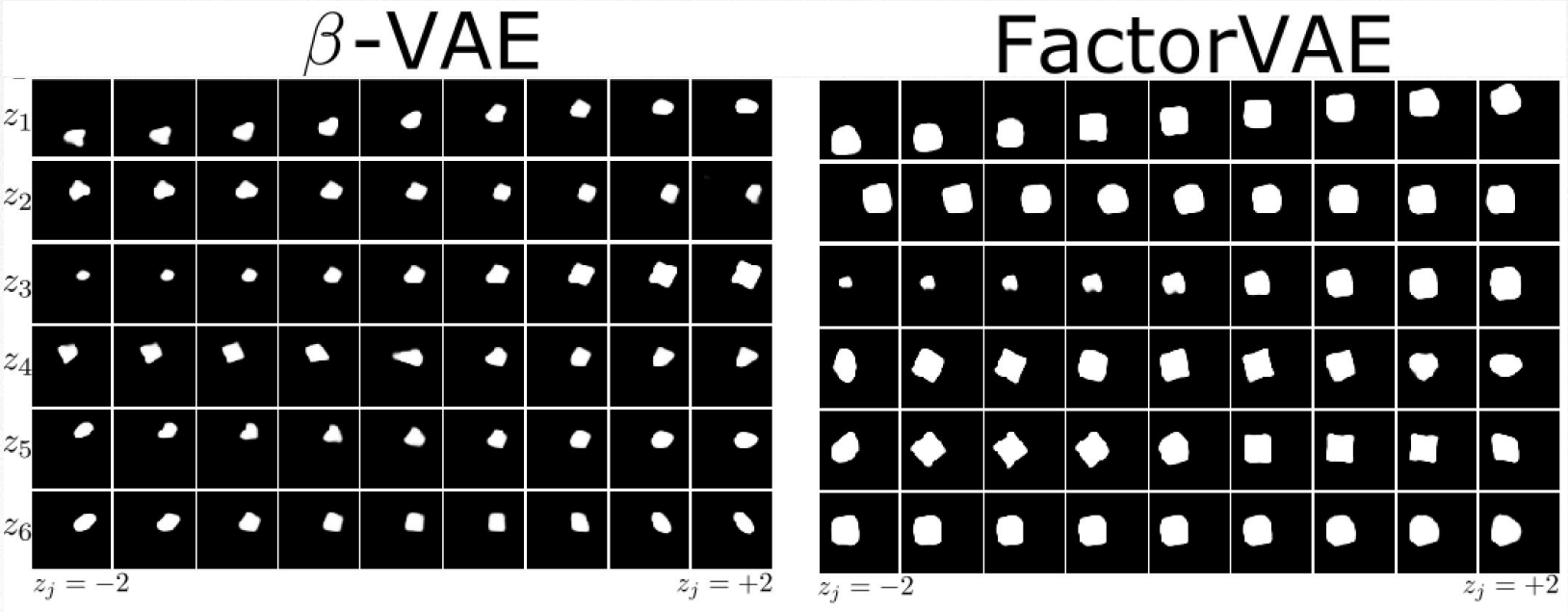
Estimating the Total Correlation

Q: How do we optimise the TC?

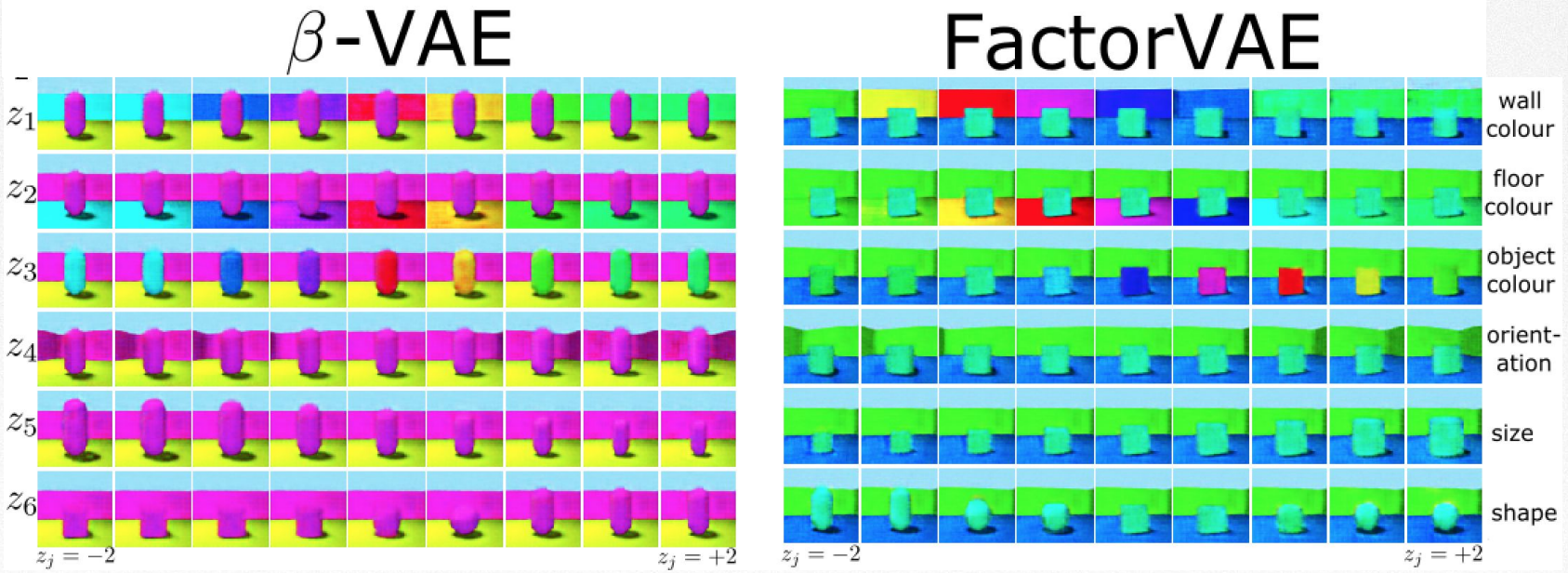
A: Use density ratio trick, using discriminator D & samples from $q(z)$

$$KL(q(z)||\prod_{j=1}^{D} q(z_j)) = \mathbb{E}_{q(z)}\left[\log \frac{q(z)}{\prod_j q(z_j)}\right] \approx \mathbb{E}_{q(z)}\left[\log \frac{D(z)}{1-D(z)}\right]$$

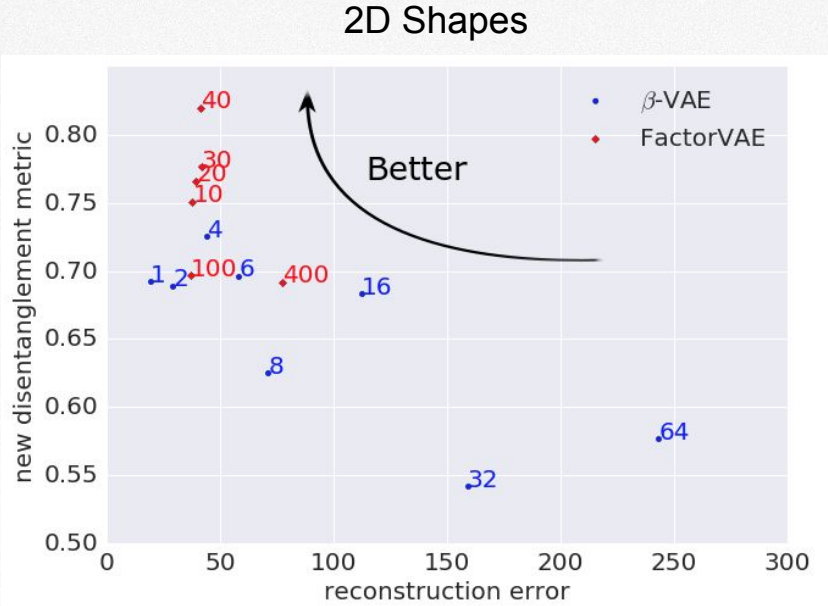
Latent Traversals (2D Shapes)

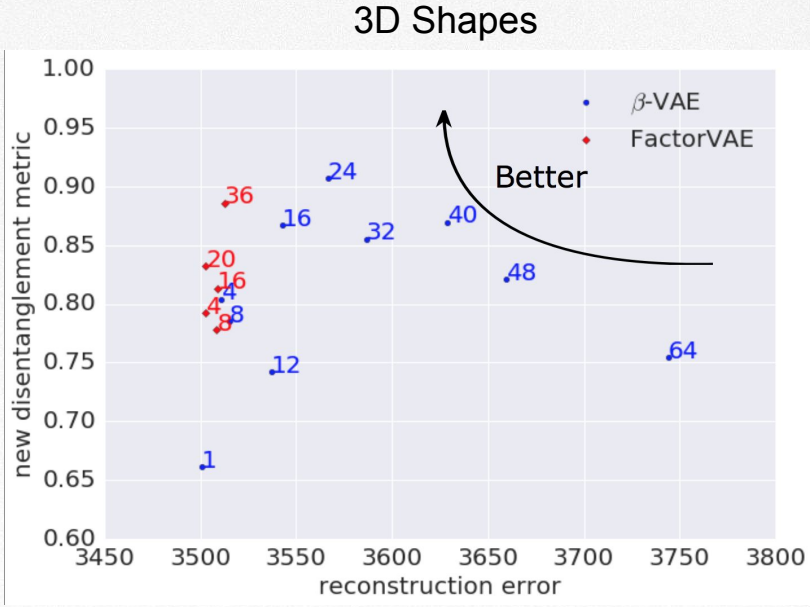


Latent Traversals (3D Shapes)



Disentanglement Reconstruction Tradeoff





DeepMind

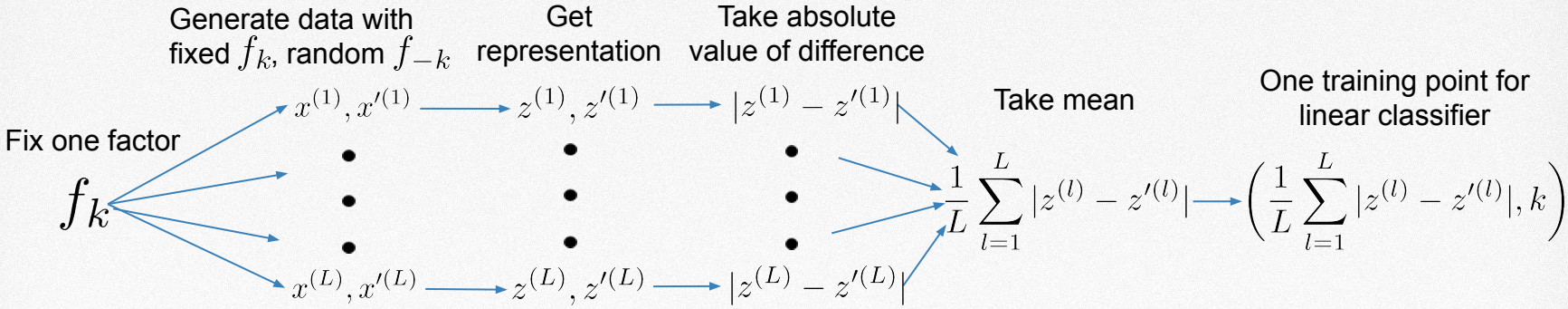
Interpretability of FactorVAE

- **Transparency**: (humans) can understand the mechanism of model/algorithm
  - **simulatability**: easily understandable computation
  - **decomposability**: each part of model (e.g. parameters, features) admits an intuitive explanation
  - **algorithmic transparency**: easy to determine whether the model will or will not work on unseen data points / datasets
- **Post-hoc interpretability**: (humans) can obtain useful information about model's mechanism and/or its predictions
  - **text explanation**
  - **visualisation:** qualitative understanding of model
  - **local (per-data point) explanation**
  - **explanation by example** e.g. finding points which the model views to be similar

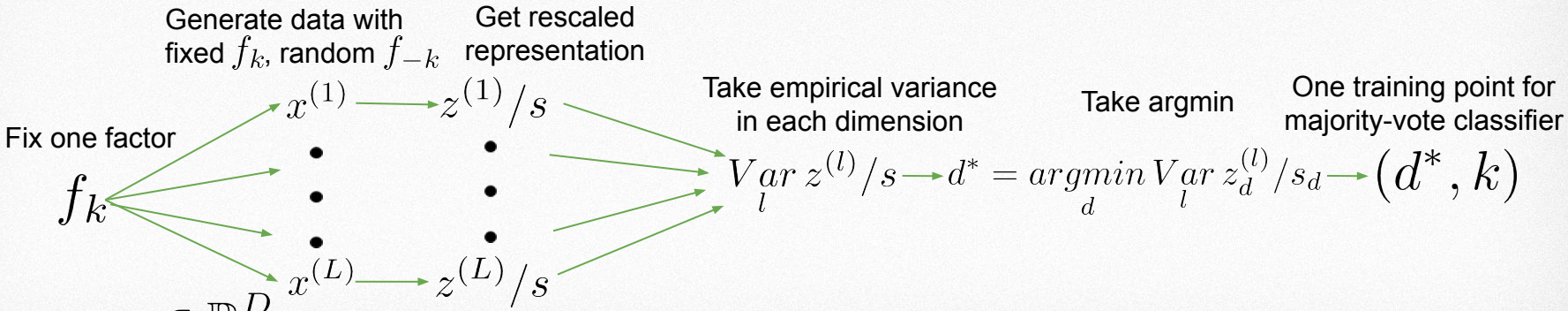
The Mythos of Model Interpretability [Lipton, '16]

Quantifying Disentanglement

- Existing disentanglement metric - supervised metric in [3]:



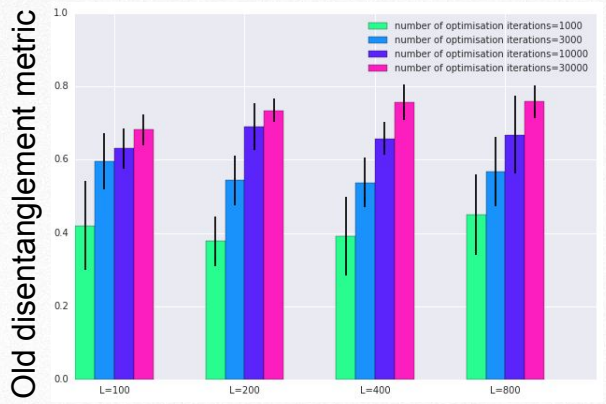
- Our improved disentanglement metric:

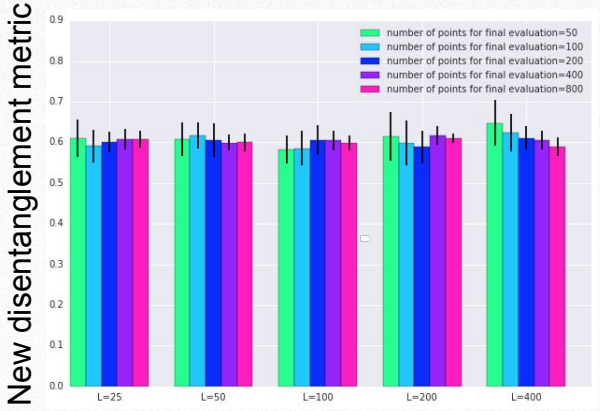


where $s \in \mathbb{R}^D$ is the scale of the latent representations of the full data (or big enough random subset)

Advantages of new disentanglement metric

- Classifier is a deterministic function of the training data (**no optimisation hyperparameters**)
- Conceptually **simpler and more intuitive** than the previous metric.
- **Circumvents failure mode** in previous metric, where it gives 100% accuracy when it only disentangles K-1 factors out of K.





DeepMind

Summary

- We introduced **FactorVAE** that **penalises Total Correlation,** giving a **better tradeoff** between reconstruction and disentanglement **than beta-VAE**.

- We introduced a **new supervised disentanglement** metric that is **simpler, more intuitive and robust** than the existing metric.

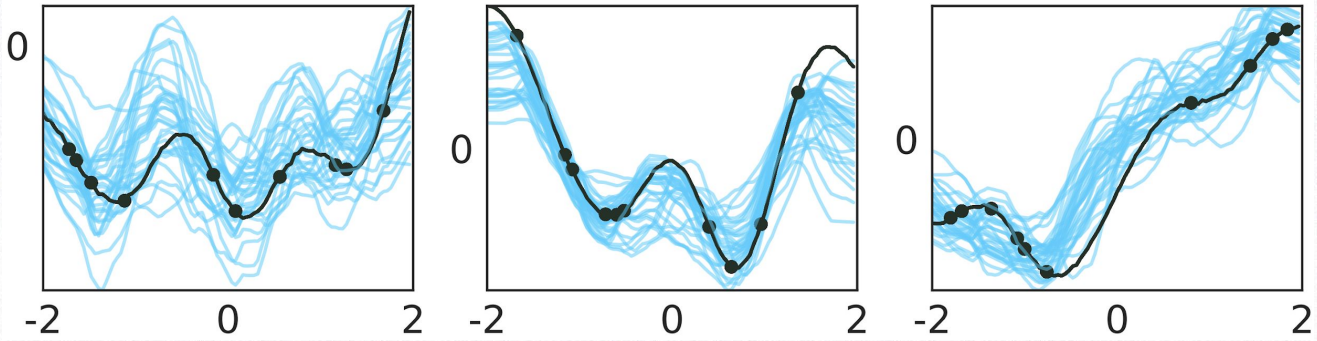
Attentive Neural Processes

To be presented @ ICLR '19

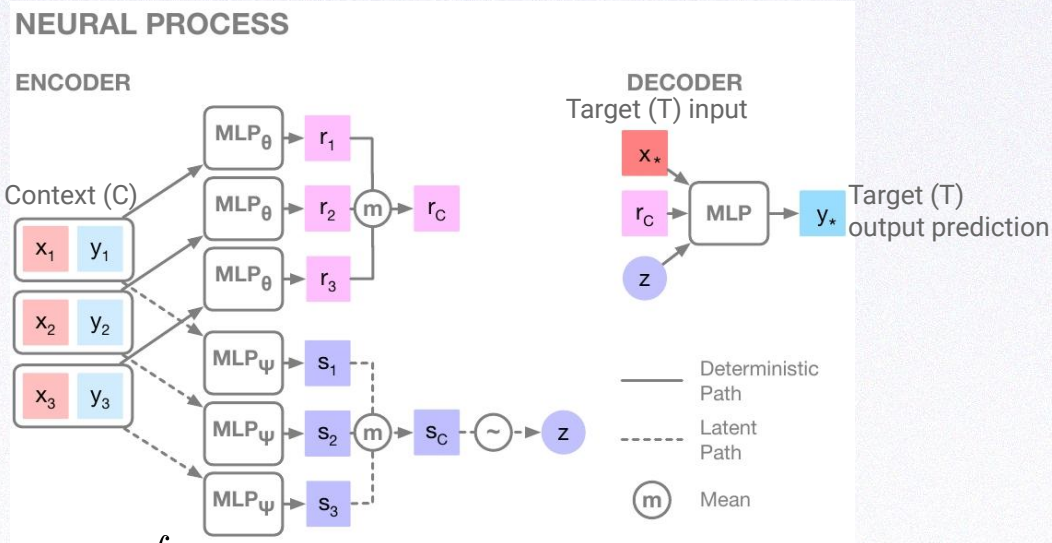
Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo,
Ali Eslami, Dan Rosenbaum, Oriol Vinyals, Yee Whye Teh

Introduction to Neural Processes (NPs)

- We explore the use of NPs for **regression.**
- Given observed $(x_i, y_i)_{i \in C}$ pairs (**context**), NPs model the function f that maps arbitrary target input x_* to the **target** output y_*.
- Specifically, **NPs learn a distribution over functions** f (i.e. stochastic process) that can explain the context data well while also giving accurate predictions on arbitrary target inputs.

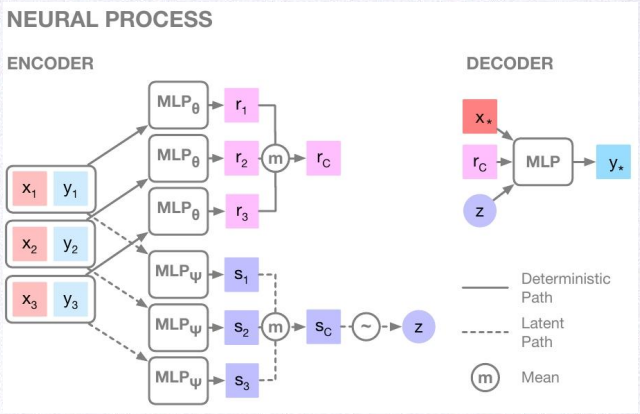


NPs



- Define: $p(\boldsymbol{y}_T|\boldsymbol{x}_T, \boldsymbol{x}_C, \boldsymbol{y}_C) := \int p(\boldsymbol{y}_T|\boldsymbol{x}_T, \boldsymbol{r}_C, \boldsymbol{z}) q(\boldsymbol{z}|\boldsymbol{s}_C) d\boldsymbol{z}$
- Learn by optimising: $\log p(\boldsymbol{y}_T|\boldsymbol{x}_T, \boldsymbol{x}_C, \boldsymbol{y}_C) \geq \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{s}_T)}[\log p(\boldsymbol{y}_T|\boldsymbol{x}_T, \boldsymbol{r}_C, \boldsymbol{z})] - D_{\mathrm{KL}}(q(\boldsymbol{z}|\boldsymbol{s}_T)\|q(\boldsymbol{z}|\boldsymbol{s}_C))$ with randomly chosen $C \subset T$

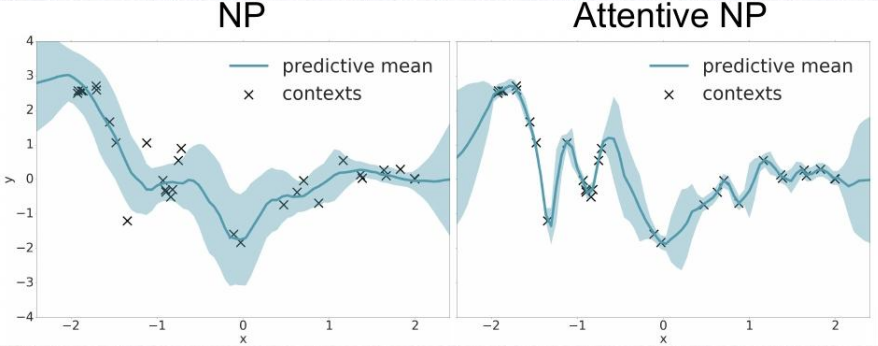
Desirable Properties of NPs

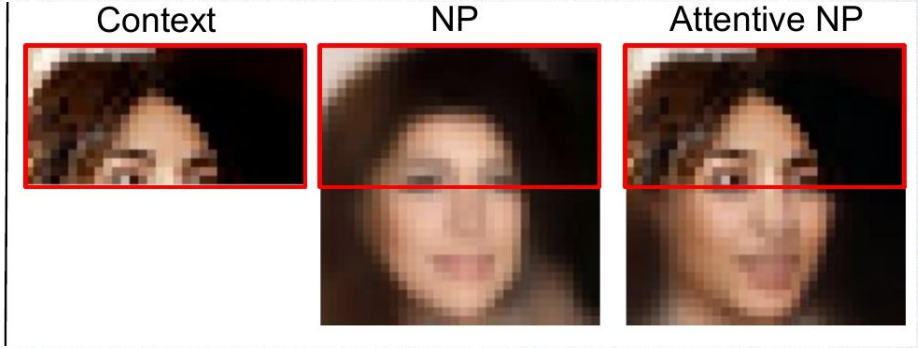


- **Linear scaling: O(*n*+*m*)** for *n* contexts and *m* targets at train and prediction time
- **Flexibility:** defines a very wide family of distributions, where one **can condition on an arbitrary number of contexts** to predict an arbitrary number of targets.
- **Order invariant** in the context points (due to aggregation of r_i by taking mean)

Problems of NPs

- Signs of **underfitting** in NPs: **inaccurate predictions at inputs of the context**
- **mean-aggregation step in encoder acts as a bottleneck**
  - **Same weight given to each context point**, so difficult for decoder to learn which contexts are relevant for given target prediction.





DeepMind

Desirable properties of GPs

- Kernel tells you which context points x_i are relevant for a given target point x_*
  - $x_* \approx x_i \Rightarrow \mathbb{E}[y_*] \approx y_i$, $\mathbb{V}[y_*] \approx 0$
  - x_* far from all $x_i \Rightarrow \mathbb{E}[y_*] \approx$ prior mean, $\mathbb{V}[y_*] \approx$ prior var
  - i.e. no risk of underfitting.

- In the land of Deep Learning, we can use differentiable ***Attention*** that **learns to attend to contexts relevant to given target**

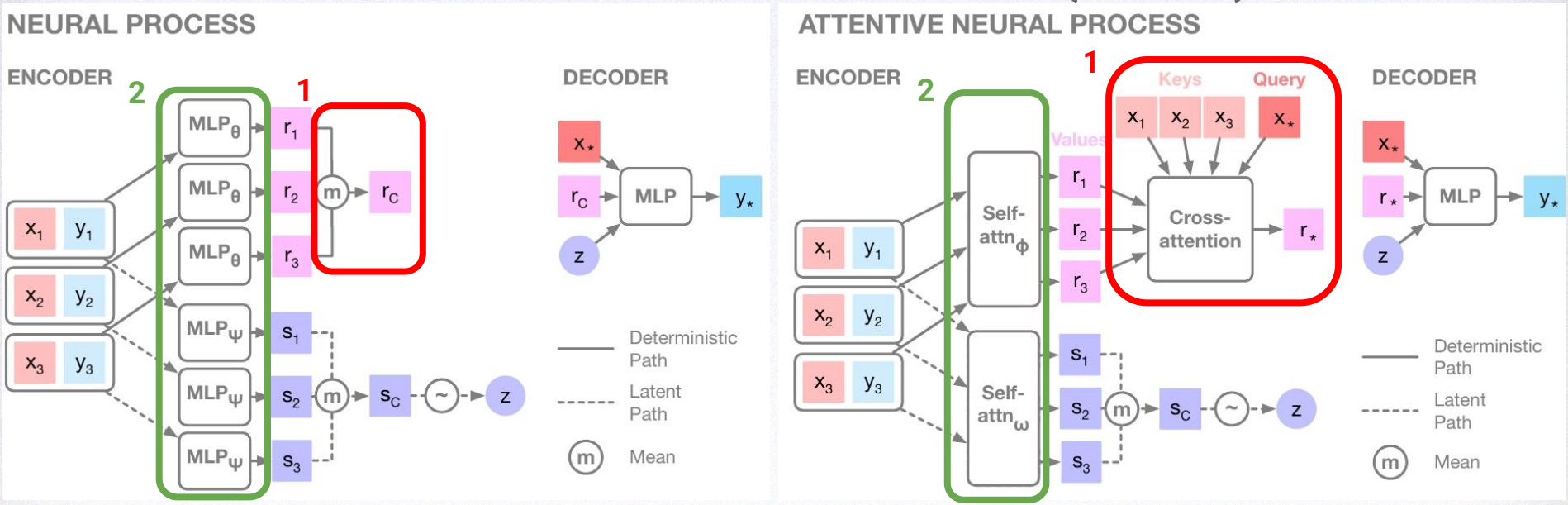
Attention

- Attention is used when we want to map query x_* and a set of key-value pairs $(x_i, y_i)_{i \in O}$ to output y_*
- It learns which (x_i, y_i) are relevant for the given x_*, which is ultimately what we want the NP to learn.
- To help NP learn this, we can **bake into NP an attention mechanism,** and this inductive bias may e.g. help avoid underfitting, enhance expressiveness of NPs, and help it learn faster.

Types of Attention

- **Laplace**: $(w_i)_{i \in C} = softmax[(-||x_i - x_*||_1)_{i \in C}], \quad r_* = \sum_{i \in C} w_i r_i$
- **Dot product**: $(w_i)_{i \in C} = softmax[(\frac{f_\theta(x_i)^\top f_\theta(x_*)}{\sqrt{d}})_{i \in C}], \quad r_*^\theta = \sum_{i \in C} w_i r_i$
  where $f_\theta = MLP_\theta, \quad d = dim(f_\theta(x))$
- **Multihead**: $r_* = Linear(Concat([r_*^{\theta_1}, \ldots, r_*^{\theta_H}]))$

Attentive Neural Processes (ANPs)

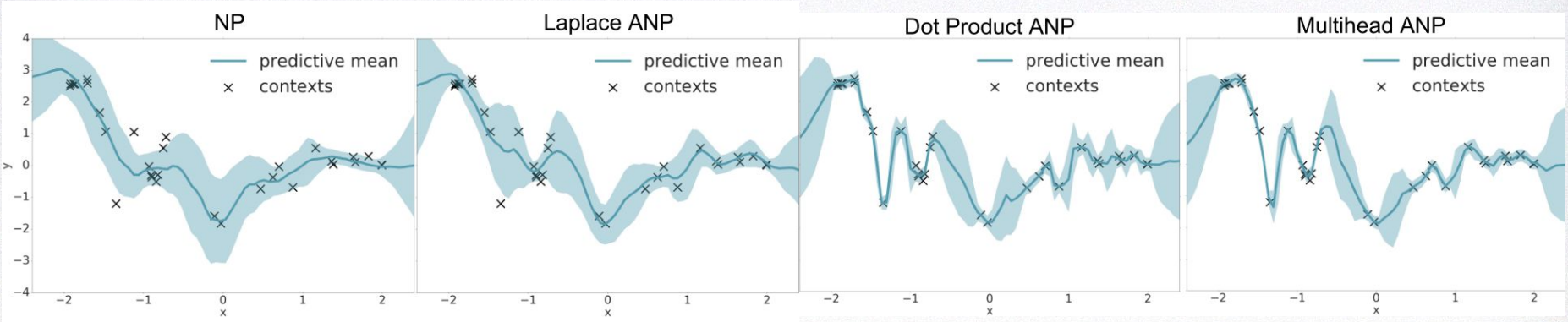


- Computational complexity risen to O(*n(n+m)*) but still fast using mini-batch training.

DeepMind

1D Function regression on GP data

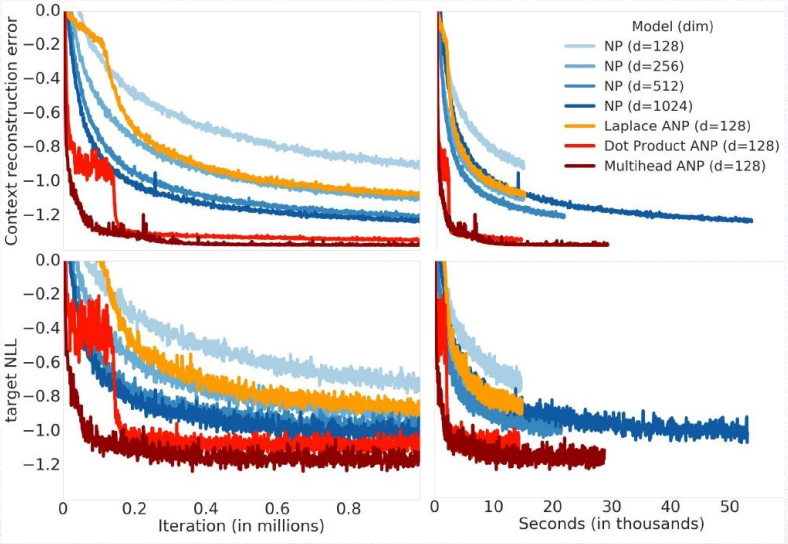
- At every training iteration, draw curve from a GP with random kernel hyperparameters (that change at every iteration).
- Then choose random points on this curve as context and targets, and optimise mini-batch loss

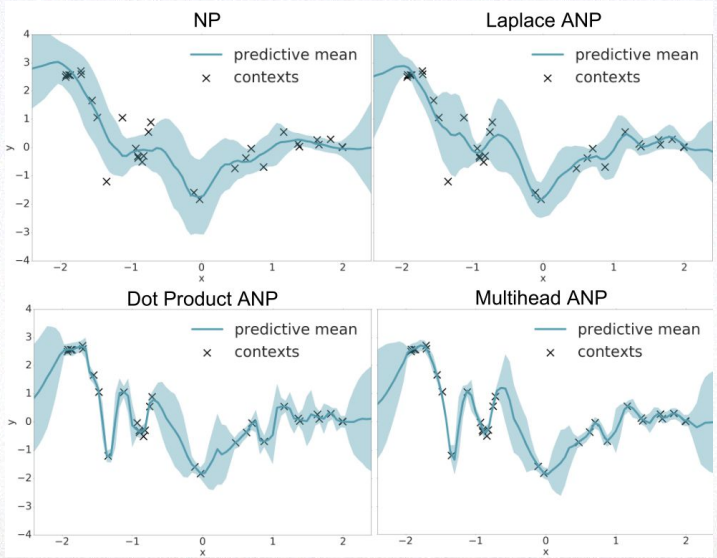


DeepMind

1D Function regression on GP data

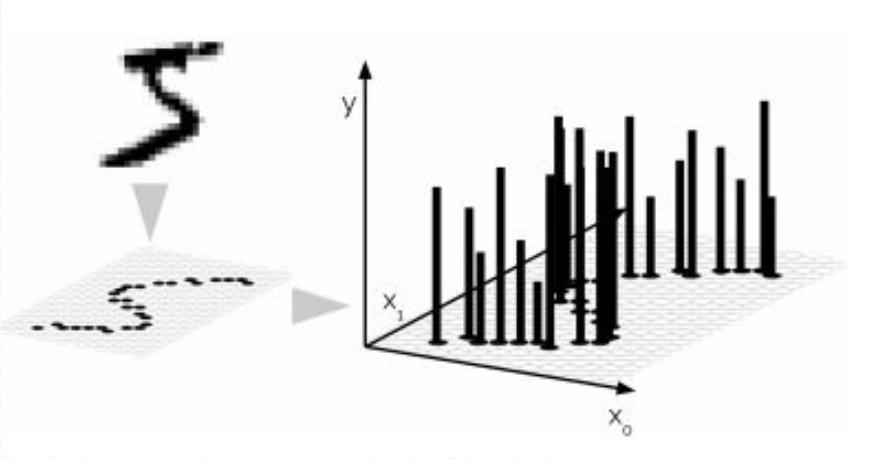
- At every training iteration, draw curve from a GP with random kernel hyperparameters (that change at every iteration).
- Then choose random points on this curve as context and targets, and optimise mini-batch loss

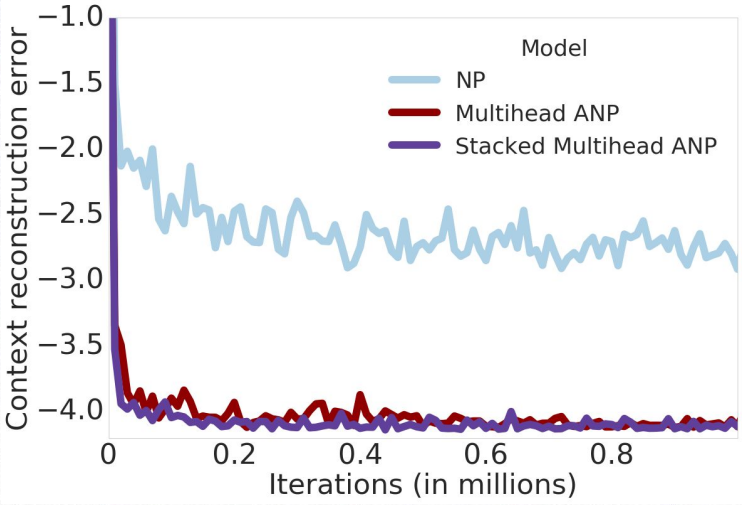




2D Function Regression on Image data

- x_i: 2D pixel coordinate, y_i: pixel intensity (1d for greyscale, 3d for RGB)
- At each training iteration, draw a random image and choose random pixels to be context and target, and optimise mini-batch loss.

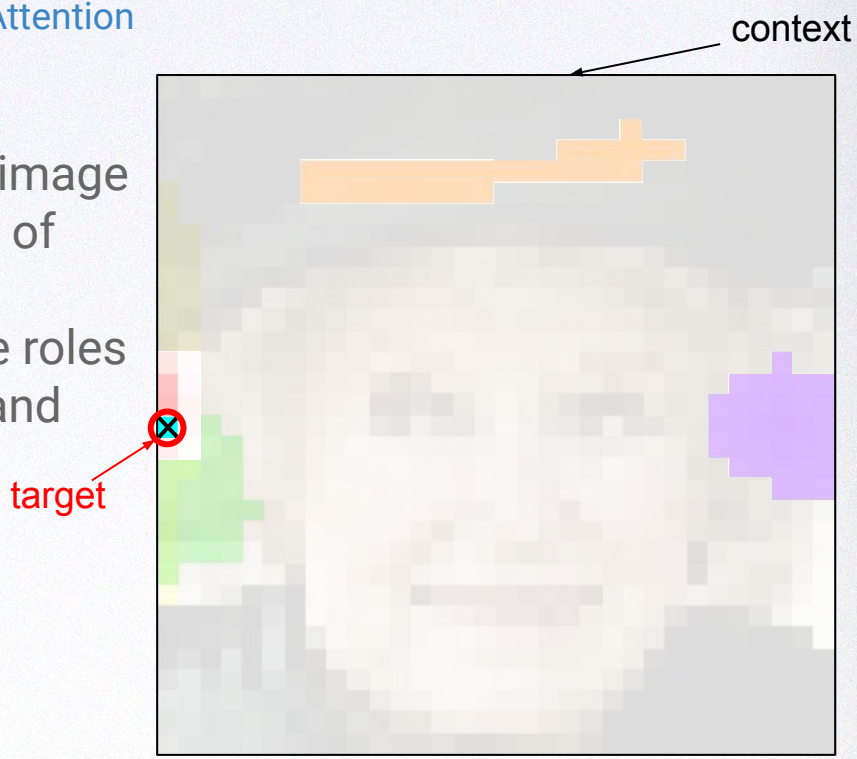




2D Function Regression on Image data

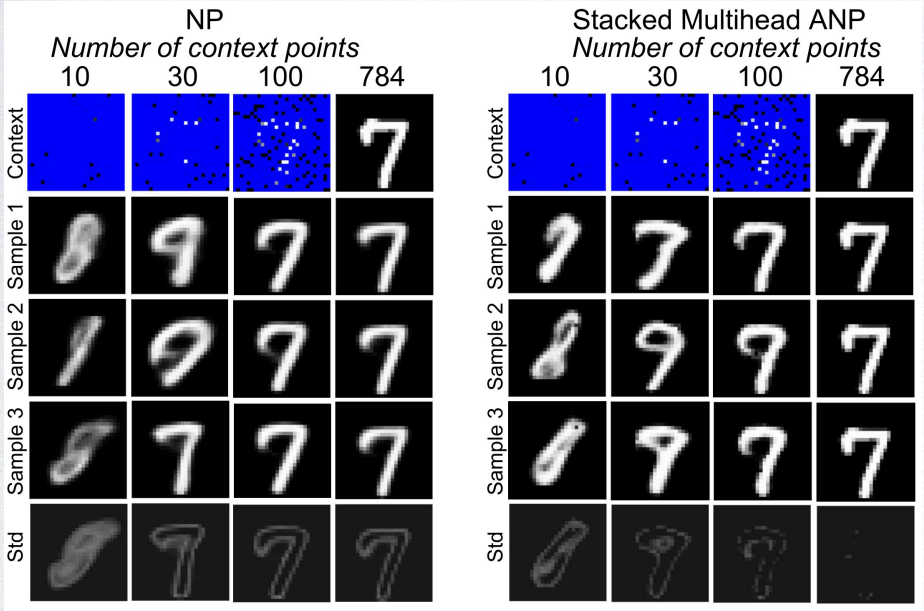
Visualisation of Attention

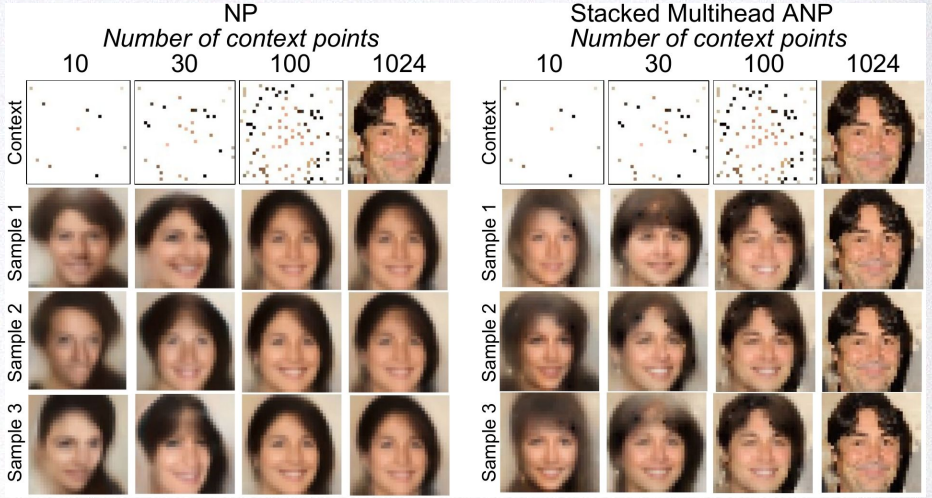
- Visualisation of Multihead Attention:
- Target is pixel with cross, context is full image
- Each **colour corresponds to** the weights of **one head of attention**.
- **Each head has different roles**, and these roles are consistent across different images and different context points.



2D Function Regression on Image data

Arbitrary Pixel Inpainting



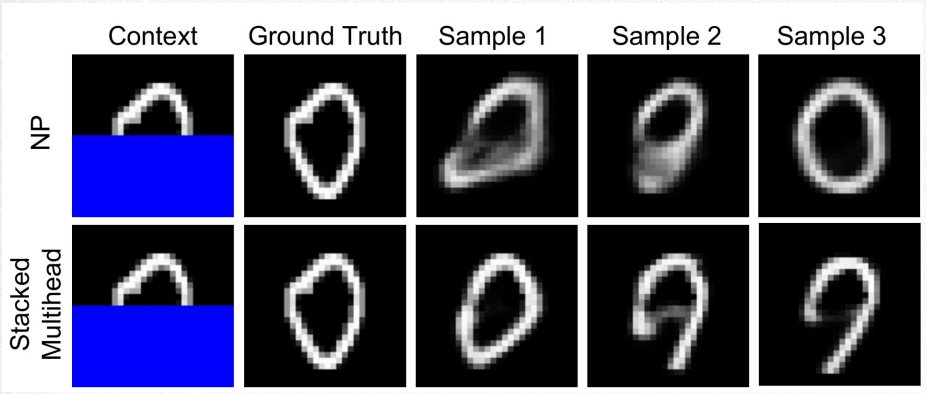


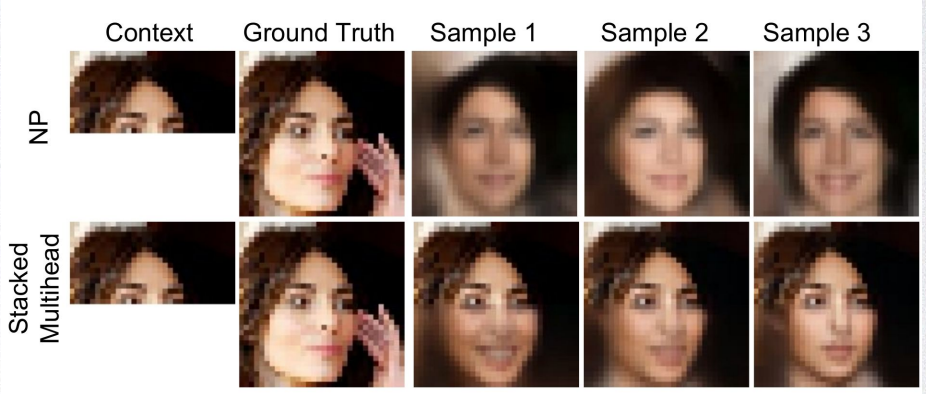
DeepMind

2D Function Regression on Image data

Bottom half prediction

Using **same model** as previous slide (with **same parameter values**):

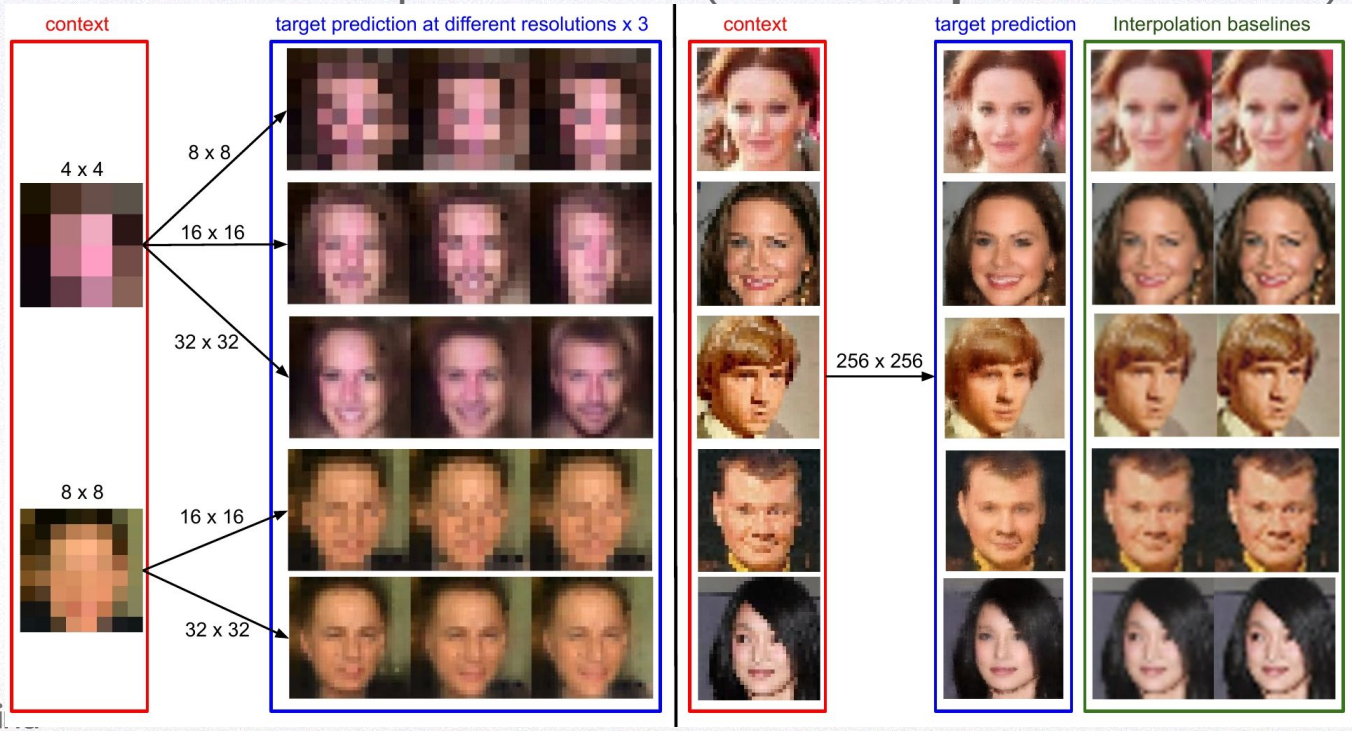




2D Function Regression on Image data

Mapping between arbitrary resolutions

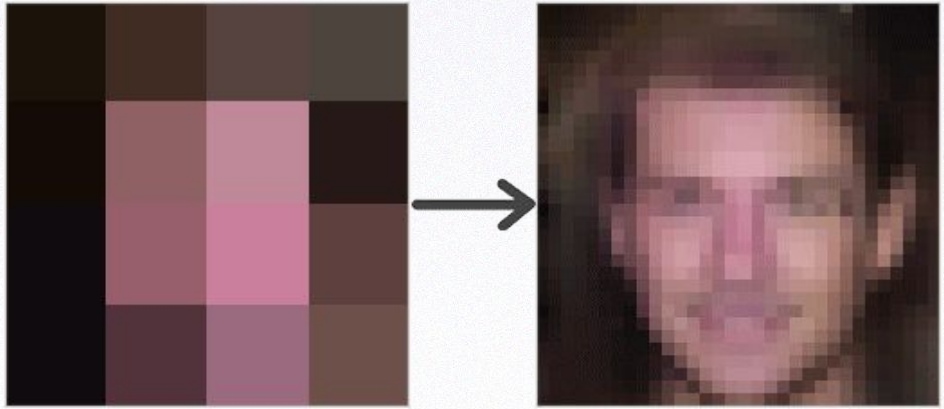
Using **same ANP model** as previous slide (with **same parameter values**):



Varying predictions with varying Latents

Bottom half prediction

Super-resolution



Interpretability of ANP

- **Transparency**: (humans) can understand the mechanism of model/algorithm
  - **simulatability**: easily understandable computation
  - **decomposability**: each part of model (e.g. parameters, features) admits an intuitive explanation
  - **algorithmic transparency**: easy to determine whether the model will or will not work on unseen data points / datasets
- **Post-hoc interpretability**: (humans) can obtain useful information about model's mechanism and/or its predictions
  - **text explanation**
  - **visualisation:** qualitative understanding of model
  - **local (per-data point) explanation**
  - **explanation by example** e.g. finding points which the model views to be similar

The Mythos of Model Interpretability [Lipton, '16]

Summary & Future Work

Compared to NPs, ANPs:

- Greatly improve the accuracy of context reconstructions and target predictions.
- Allow faster training.
- Expand the range of functions that can be modelled.

Future Work:

- Application to few-shot regression & comparison to standard Meta-Learning algorithms.
- Using self-attention in the decoder for the image application (links to Image Transformer)

Conclusions

- We've introduced FactorVAE, a model for learning disentangled representations that shows various interpretable properties.
- We've introduced ANP, a model for learning stochastic processes that also shows interpretable properties.
- Deep Learning and interpretability are compatible!
- We should think more about interpretability when devising new DL models!