Meta-learning and one-shot learning

NIPS 2016 take-aways

Olof Mogren

Chalmers University of Technology

2017-02-02

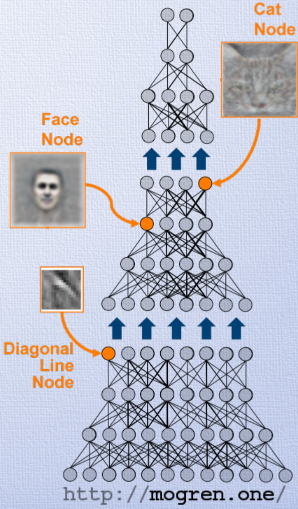
Coming seminars

- Today: Olof Mogren
  Meta-learning and one-shot learning
- February 9: Devdatt Dubhashi
- February 16: **Up for grabs! Get in touch:**
  `mogren@chalmers.se`

`http://www.cse.chalmers.se/research/lab/seminars/`

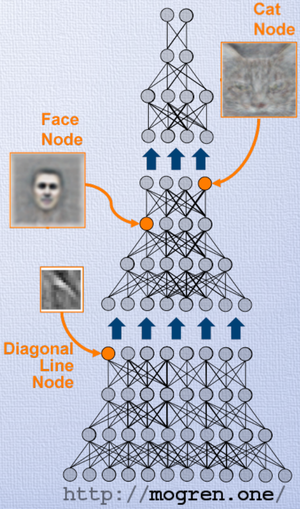
In the beginning there were features

- ...



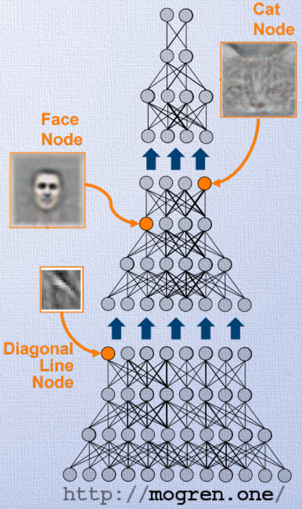
In the beginning there were features

- ...
- But they were engineered by hand



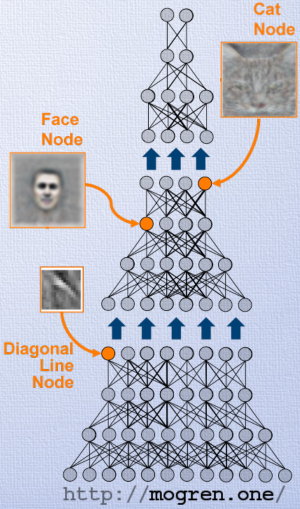
In the beginning there were features

- ...
- But they were engineered by hand
- Along came feature learning



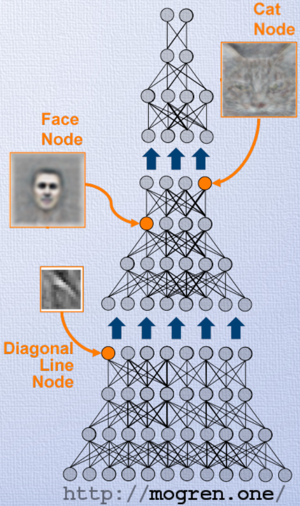
IN THE BEGINNING THERE WERE FEATURES

- ...
- But they were engineered by hand
- Along came feature learning
- Deep NNs learn a larger part of the problem



In the beginning there were features

- ...
- But they were engineered by hand
- Along came feature learning
- Deep NNs learn a larger part of the problem
- Can we learn more?



Meta learning

- "Learning to learn"

Meta learning

- "Learning to learn"
- Schmidthuber 1987, 1992, 1993 - nets modifying their own weights

Meta learning

- "Learning to learn"
- Schmidthuber 1987, 1992, 1993 - nets modifying their own weights
- Schmidthuber 1997 - The Success Story Algorithm

Meta learning

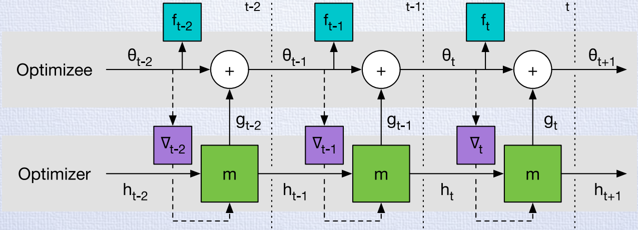
- "Learning to learn"
- Schmidthuber 1987, 1992, 1993 - nets modifying their own weights
- Schmidthuber 1997 - The Success Story Algorithm
- Daniel, et.al. 2016 - Reinforcement learning

META LEARNING

- "Learning to learn"
- Schmidthuber 1987, 1992, 1993 - nets modifying their own weights
- Schmidthuber 1997 - The Success Story Algorithm
- Daniel, et.al. 2016 - Reinforcement learning
- Santoro, et.al. 2016, **Vinyals, et.al.** 2016 - One-shot learning, mem-augmented NNs (multi-task learning is generalization)

Learning to learn by gradient descent

by gradient descent

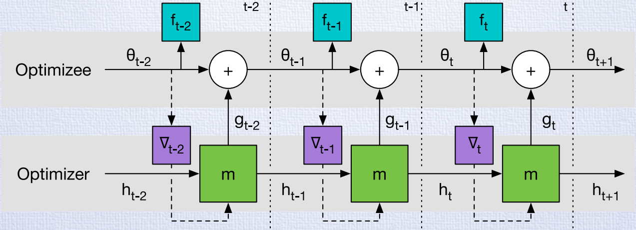


- A learned coordinate-wise optimizer

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, Nando de Freitas

Learning to learn by gradient descent

by gradient descent

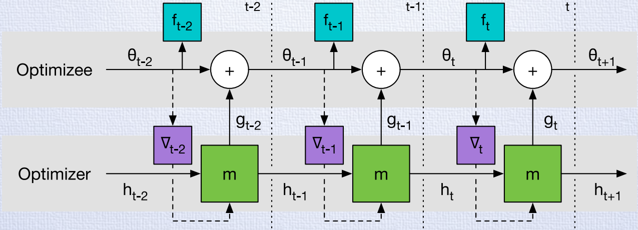


- A learned coordinate-wise optimizer
- Two-layered LSTM net

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, Nando de Freitas

Learning to learn by gradient descent

BY GRADIENT DESCENT

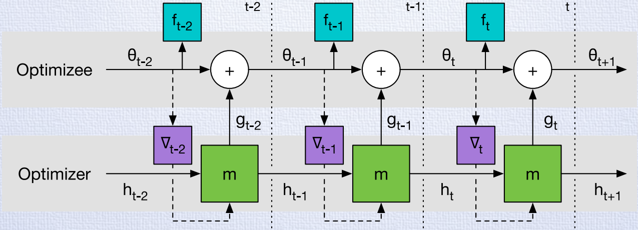


- A learned coordinate-wise optimizer
- Two-layered LSTM net
- **Inputs: optimizee** gradient for one coordinate, **optimizer** state

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, Nando de Freitas

Learning to learn by gradient descent

by gradient descent



- A learned coordinate-wise optimizer
- Two-layered LSTM net
- **Inputs: optimizee** gradient for one coordinate, **optimizer** state
- **Outputs:** update for the specific coordinate

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, Nando de Freitas

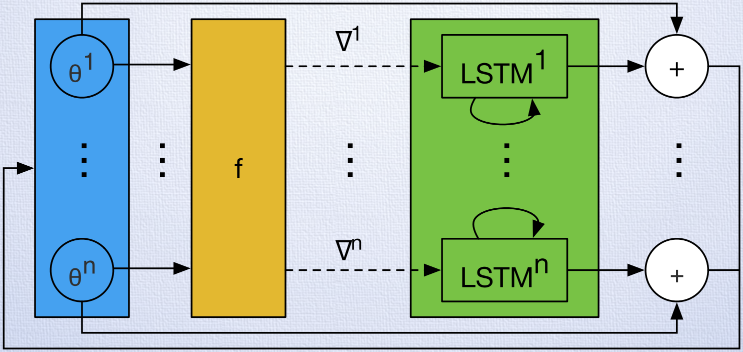
LEARNING TO LEARN BY GRADIENT DESCENT

BY GRADIENT DESCENT

$$\mathcal{L}(\phi) = \mathbb{E}_f \left[\sum_{t=1}^{T} w_t f(\theta_t) \right] \quad \text{where} \quad \begin{aligned} \theta_{t+1} &= \theta_t + g_t \,, \\ \begin{bmatrix} g_t \\ h_{t+1} \end{bmatrix} &= m(\nabla_t, h_t, \phi) \,. \end{aligned} \tag{1}$$

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, Nando de Freitas

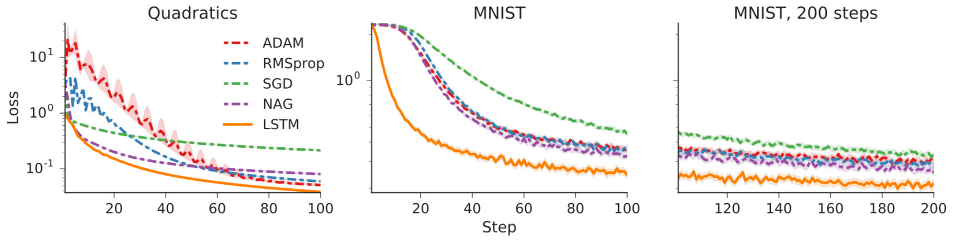
Coordinate-wise LSTM optimizer



$LSTM^1 \ldots LSTM^n$ have shared weights, but separate hidden states.

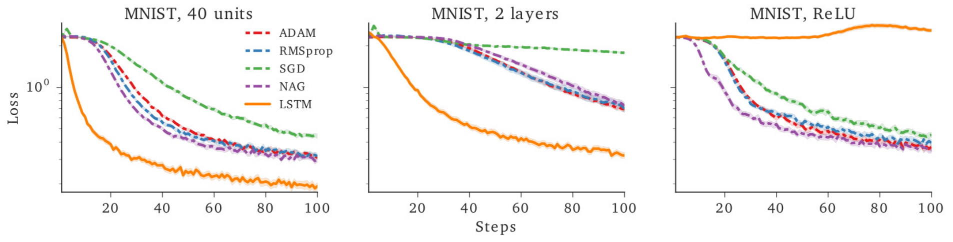
Andrychowicz, et.al., Learning to learn by gradient descent by gradient descent (NIPS 2016)

Learning curves



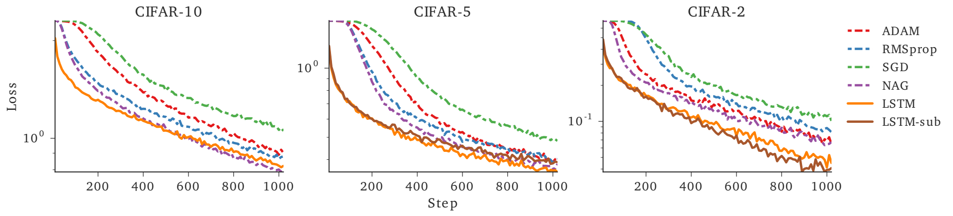
Andrychowicz, et.al., Learning to learn by gradient descent by gradient descent (NIPS 2016)

GENERALIZATION, MODEL LAYOUT



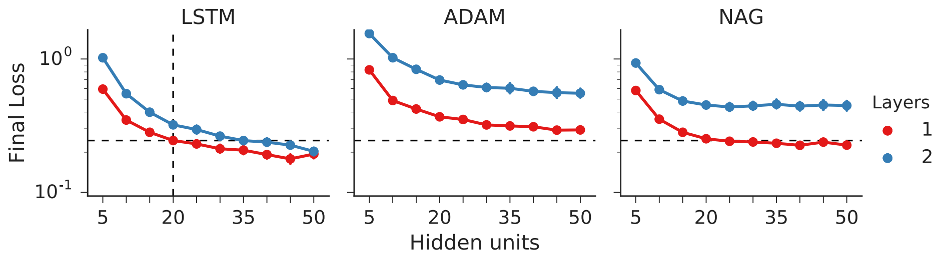
Andrychowicz, et.al., Learning to learn by gradient descent by gradient descent (NIPS 2016)

Generalization, CIFAR-10/5/2



Andrychowicz, et.al., Learning to learn by gradient descent by gradient descent (NIPS 2016)

Generalization, model size



Andrychowicz, et.al., Learning to learn by gradient descent by gradient descent (NIPS 2016)

So...

- Meta-learning allows us to learn how to learn

So...

- Meta-learning allows us to learn how to learn
- Generalize to new problems

One-shot learning

- How do we learn when we have very limited data?

One-shot learning

- How do we learn when we have very limited data?
- Exampe: k-nearest neighbours (no learning/zero-shot)

One-shot learning

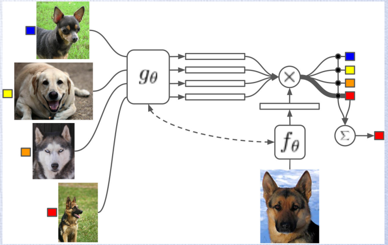
- How do we learn when we have very limited data?
- Exampe: k-nearest neighbours (no learning/zero-shot)
- One-shot learning: learn from few examples

One-shot learning

- How do we learn when we have very limited data?
- Exampe: k-nearest neighbours (no learning/zero-shot)
- One-shot learning: learn from few examples
- Also meta-learning; learn from one large training set how to make use of smaller data.

Matching networks for one-shot learning

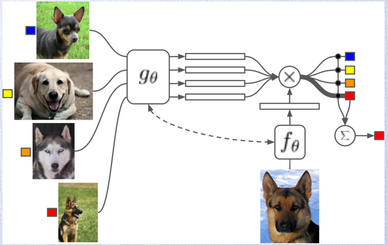
- Related to metric learning



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

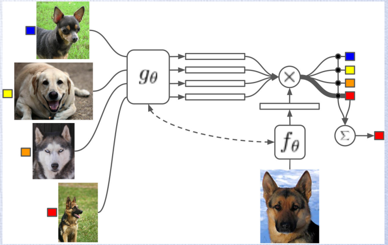
- Related to metric learning
- Deep neural features



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

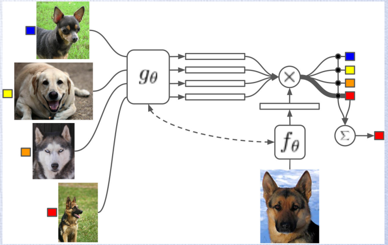
- Related to metric learning
- Deep neural features
- Small labelled support set S,



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

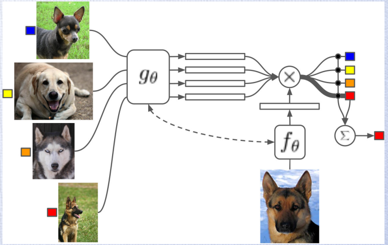
- Related to metric learning
- Deep neural features
- Small labelled support set S,
- Larns to map S to a cassifier $c(x)$.



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

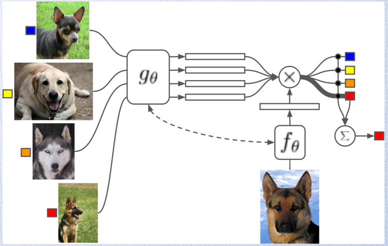
- Related to metric learning
- Deep neural features
- Small labelled support set *S*,
- Larns to map *S* to a cassifier *c*(*x*).
- *S* may contain unseen classes!



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

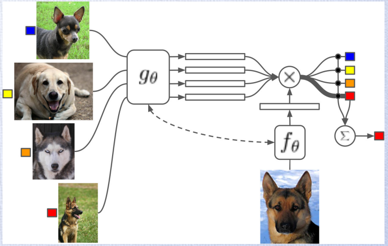
- $S = (x_i, y_i)_{i=1}^{k}$



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

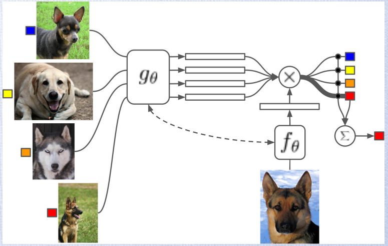
- $S = (x_i, y_i)_{i=1}^{k}$
- $\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

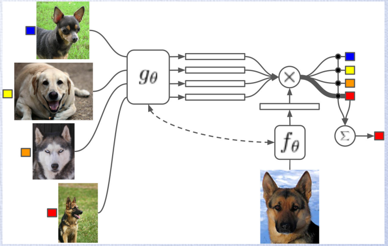
- $S = (x_i, y_i)_{i=1}^{k}$
- $\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$
- (If a is a kernel, then this is a kernel density estimator)



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

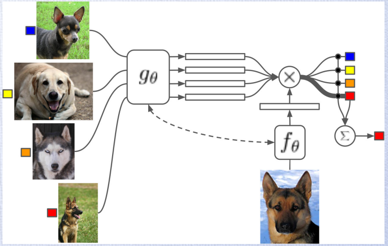
- $S = (x_i, y_i)_{i=1}^{k}$
- $\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$
- (If a is a kernel, then this is a kernel density estimator)
- Subsumes both KDE and kNN



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

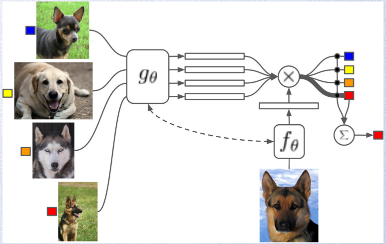
- $S = (x_i, y_i)_{i=1}^{k}$
- $\hat{y} = \sum_{i=1}^{k} a(\hat{x}, x_i) y_i$
- (If a is a kernel, then this is a kernel density estimator)
- Subsumes both KDE and kNN
- $a(\hat{x}, x_i) = \frac{e^{c(f(\hat{x}), g(x_i))}}{Z}$
  (Softmax of cosine sim)



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

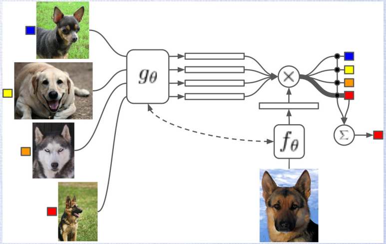
- $a(\hat{x}, x_i) = \frac{e^{\cos(f(\hat{x}), g(x_i))}}{Z}$
  (Softmax of cosine sim)



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

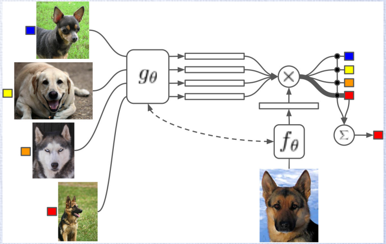
- $a(\hat{x}, x_i) = \frac{e^{\cos(f(\hat{x}), g(x_i))}}{Z}$
  (Softmax of cosine sim)
  - f (embedding of new instances)



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

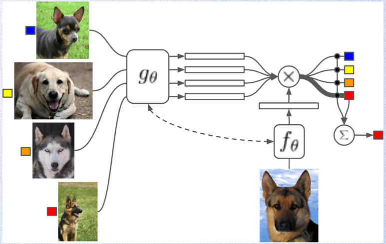
- $a(\hat{x}, x_i) = \frac{e^{\cos(f(\hat{x}), g(x_i))}}{Z}$ (Softmax of cosine sim)
    - f (embedding of new instances)
    - g (embedding of support set)



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

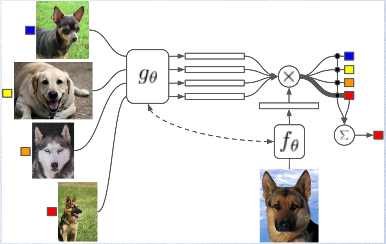
- $a(\hat{x}, x_i) = \frac{e^{\cos(f(\hat{x}), g(x_i))}}{Z}$
  (Softmax of cosine sim)
  - f (embedding of new instances)
  - g (embedding of support set)
  - VGG, Inception, word embeddings



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

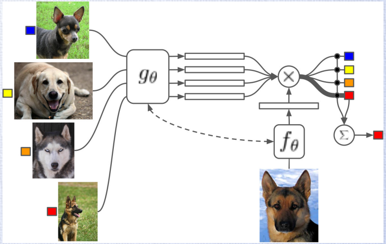
- $a(\hat{x}, x_i) = \frac{e^{\cos(f(\hat{x}), g(x_i))}}{Z}$
  (Softmax of cosine sim)
  - f (embedding of new instances)
  - g (embedding of support set)
  - VGG, Inception, word embeddings
  - $f(\hat{x}, S) = LSTM_{attention}(f'(\hat{x}), g(S), K)$



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

- $a(\hat{x}, x_i) = \frac{e^{cos(f(\hat{x}), g(x_i))}}{Z}$ (Softmax of cosine sim)
    - f (embedding of new instances)
    - g (embedding of support set)
    - VGG, Inception, word embeddings
    - $f(\hat{x}, S) = LSTM_{attention}(f'(\hat{x}), g(S), K)$

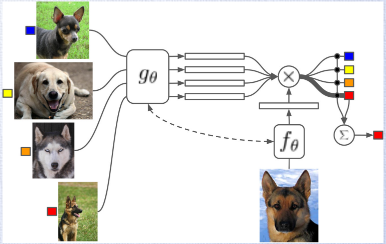


- $\theta = argmax_\theta E_{L\sim T}\left[E_{S\sim L, B\sim L}\left[\sum_{(x,y)\in B} \log P(y|x,S)\right]\right]$

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

Matching networks for one-shot learning

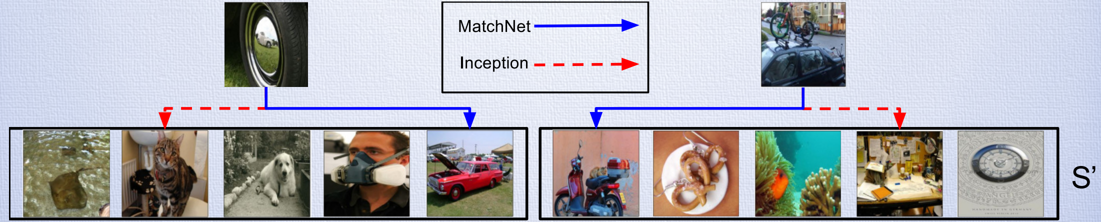
- $a(\hat{x}, x_i) = \frac{e^{cos(f(\hat{x}), g(x_i))}}{Z}$ (Softmax of cosine sim)
  - f (embedding of new instances)
  - g (embedding of support set)
  - VGG, Inception, word embeddings
  - $f(\hat{x}, S) = LSTM_{attention}(f'(\hat{x}), g(S), K)$



- $\theta = argmax_\theta E_{L \sim T}\left[E_{S \sim L, B \sim L}\left[\sum_{(x,y) \in B} \log P(y|x, S)\right]\right]$
- (learns to learn from a support set).

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

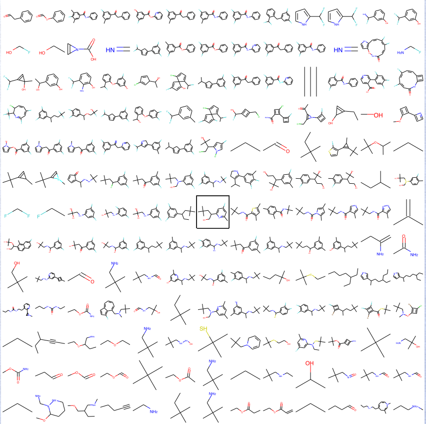
Matching networks: examples



Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, Daan Wierstra (NIPS 2016)

AUTOMATIC CHEMICAL DESIGN

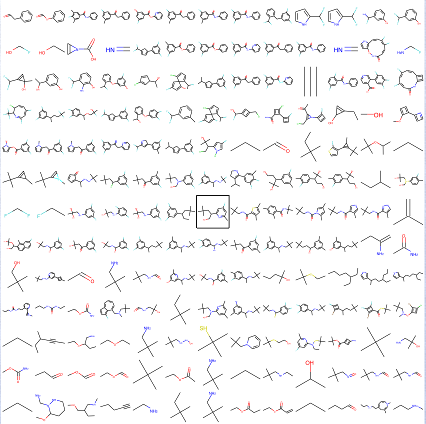
- Search in molecular space is challenging; large, discrete, and unstructured



Rafael Gómez-Bombarelli, David Duvenaud, José Miguel Hernández-Lobato, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, Alán Aspuru-Guzik, 2016

Automatic chemical design

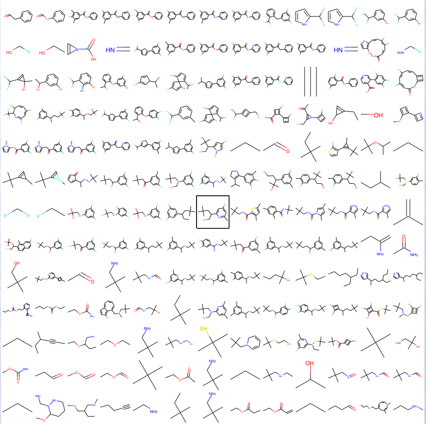
- Search in molecular space is challenging; large, discrete, and unstructured
- Variational autoencoder



Rafael Gómez-Bombarelli, David Duvenaud, José Miguel Hernández-Lobato, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, Alán Aspuru-Guzik, 2016

AUTOMATIC CHEMICAL DESIGN

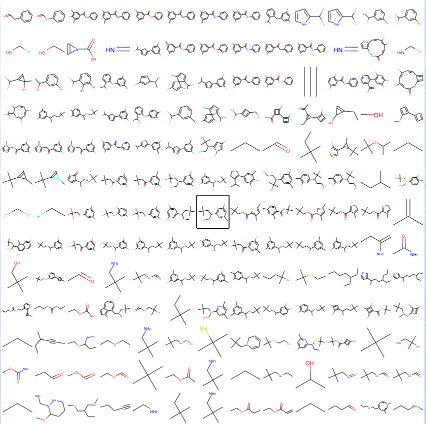
- Search in molecular space is challenging; large, discrete, and unstructured
- Variational autoencoder
- Convert discrete representations to and from continuous



Rafael Gómez-Bombarelli, David Duvenaud, José Miguel Hernández-Lobato, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, Alán Aspuru-Guzik, 2016

AUTOMATIC CHEMICAL DESIGN

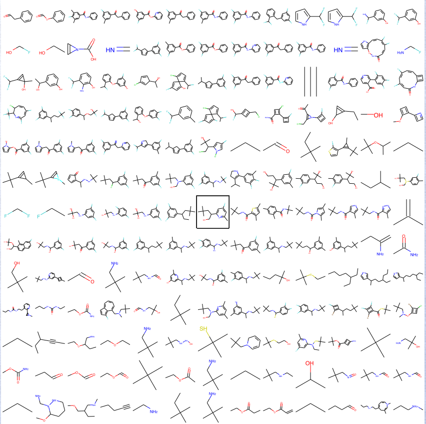
- Search in molecular space is challenging; large, discrete, and unstructured
- Variational autoencoder
- Convert discrete representations to and from continuous
- Optimize molecule properties



Rafael Gómez-Bombarelli, David Duvenaud, José Miguel Hernández-Lobato, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, Alán Aspuru-Guzik, 2016

Automatic chemical design

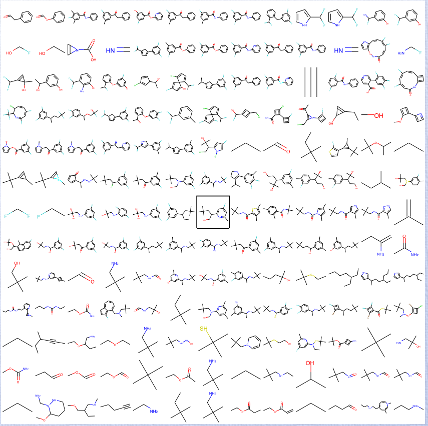
- Search in molecular space is challenging; large, discrete, and unstructured
- Variational autoencoder
- Convert discrete representations to and from continuous
- Optimize molecule properties
- Best paper award at Constructive machine learning workshop at NIPS



Rafael Gómez-Bombarelli, David Duvenaud, José Miguel Hernández-Lobato, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, Alán Aspuru-Guzik, 2016

Automatic chemical design

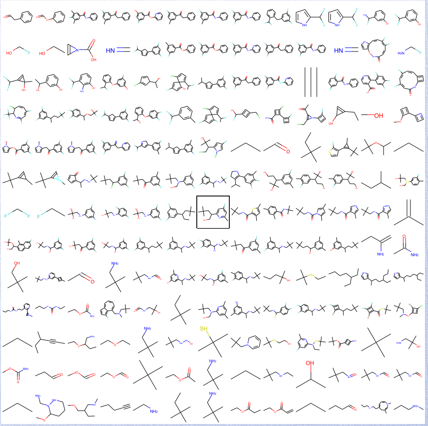
- Variational autoencoder, seq2seq



Gómez-Bombarelli, et.al., 2016

AUTOMATIC CHEMICAL DESIGN

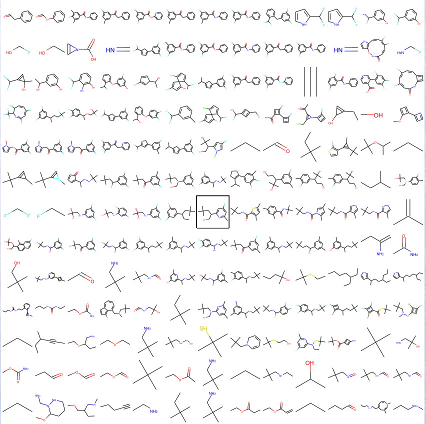
- Variational autoencoder, seq2seq
- String representation of molecules: SMILES



Gómez-Bombarelli, et.al., 2016

Automatic chemical design

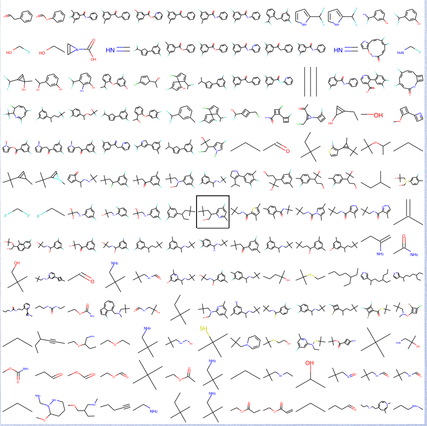
- Variational autoencoder, seq2seq
- String representation of molecules: SMILES
- Decode random vectors



Gómez-Bombarelli, et.al., 2016

Automatic chemical design

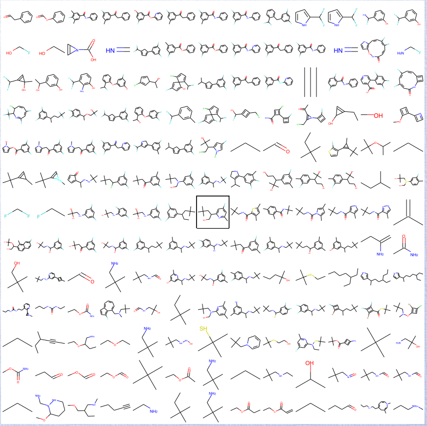
- Variational autoencoder, seq2seq
- String representation of molecules: SMILES
- Decode random vectors
- Perturb known chemical structures



Gómez-Bombarelli, et.al., 2016

Automatic chemical design

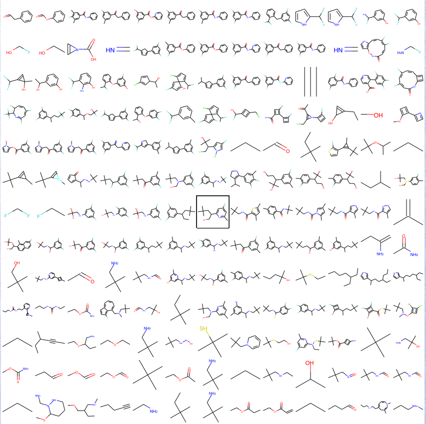
- Variational autoencoder, seq2seq
- String representation of molecules: SMILES
- Decode random vectors
- Perturb known chemical structures
- Interpolate between molecules



Gómez-Bombarelli, et.al., 2016

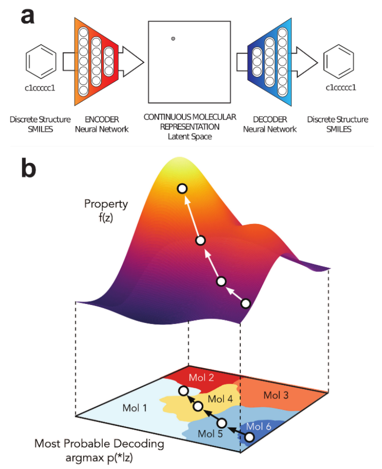
AUTOMATIC CHEMICAL DESIGN

- Variational autoencoder, seq2seq
- String representation of molecules: SMILES
- Decode random vectors
- Perturb known chemical structures
- Interpolate between molecules
- Train a model to predict medical properties based on representation



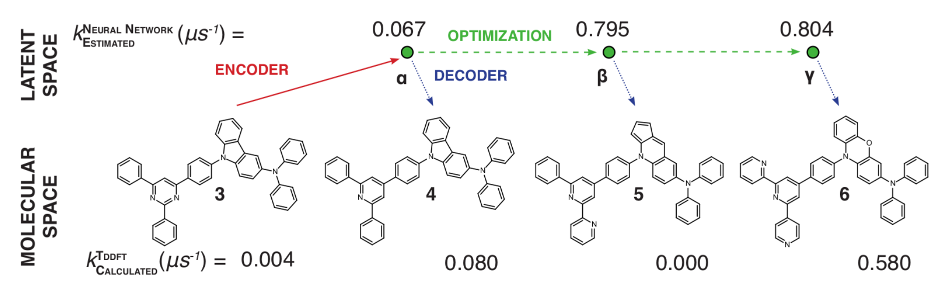
Gómez-Bombarelli, et.al., 2016

Automatic chemical design



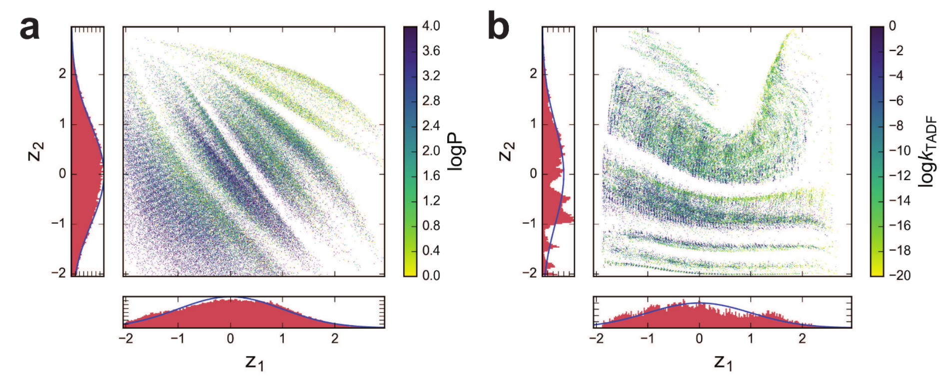
Gómez-Bombarelli, et.al., 2016

AUTOMATIC CHEMICAL DESIGN



Gómez-Bombarelli, et.al., 2016

AUTOMATIC CHEMICAL DESIGN



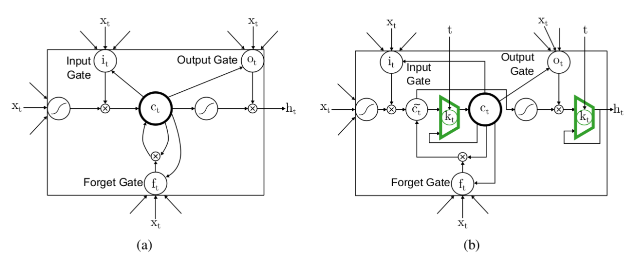
Gómez-Bombarelli, et.al., 2016

mogren@chalmers.se

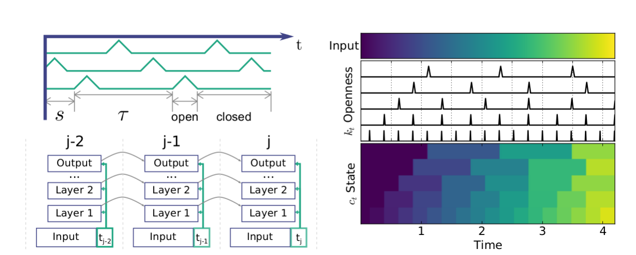
http://mogren.one/

APPENDIX

PHASED LSTM



Phased LSTM: state visualization



http://mogren.one/