CS273B
Fall 2016

Main Takeaways/Final Remarks

Focus Areas

Functional Genomics

- Sequence variants
- Population genomics

DeepSEA

DeMo Dashboard

Basset

DanQ

DeepBind

DeepCpG

Pharmacogenomics

- Small molecule drug candidates
- Subcellular protein localization
- Protein structure prediction

Convolutional LSTM networks

DeepCNF AtomNet

ResNets for contact maps

Molecular graph convolutions

Medical records/clincal data

DeepPatient

DeepCare

Deep survival analysis

Medical Imaging

3D CNN's with Fully Connected CRF's

DeepCyTOF

Algorithms from computer vision:
VGGNet, GoogleNet

Lesson 0: Think very deeply about the problem formulation, evaluation, and data processing.

Problem formulation and data pre-processing

- In supervised models: how to select background/negative sets.
  - It matters a lot or you can fool yourself into having a good model
  - Adaptive/iterative sampling of negative background (think of boosting)

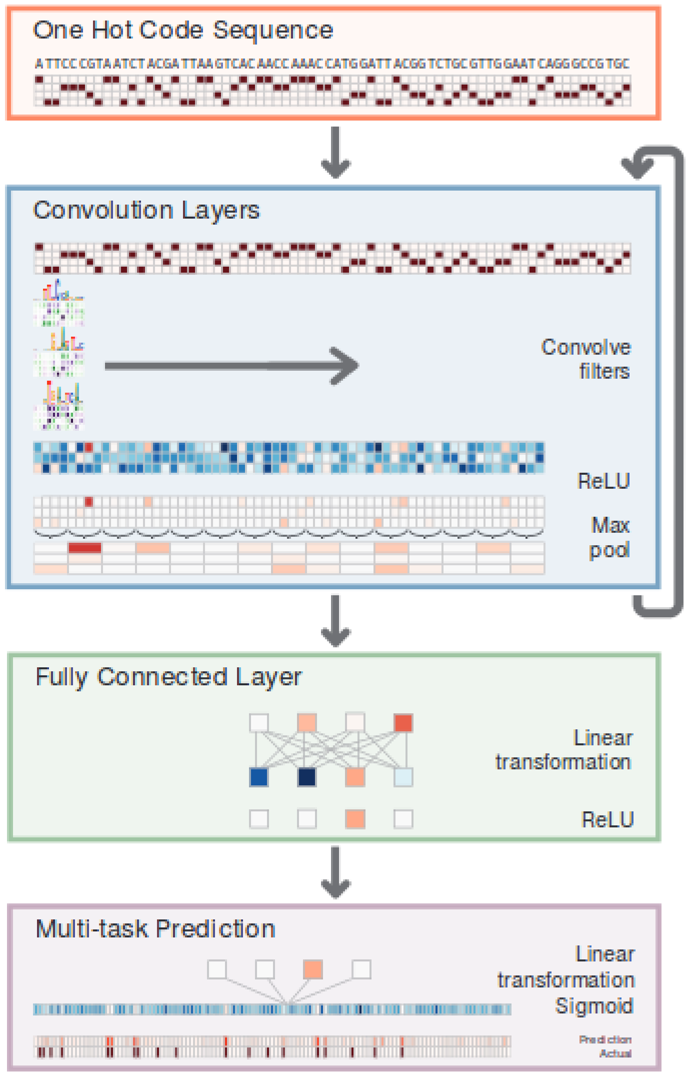
- Be very aware of data processing and data artifacts unique to measurement technologies and confounding factors in biology.
  - This is far less common in classical application areas of deep learning such as vision, speech and NLP.
- Domain adaption and domain adversarial training
  - Automatic data normalization methods based on encoder-decoder
  - Encode low sequencing depth data to decode into high sequencing depth data – let the neural net learn to correct for confounder lab effect

- Data augmentation strategies specific to genomics
- Subsampling sequencing, corrupting in biologically meaningful ways

- Think of ways to use prior knowledge
- Architectures can often be well motivated by biology
- Initializations can be well motivated (think DanQ paper)

Lesson 1: Match the right neural network architecture to the appropriate problem setup

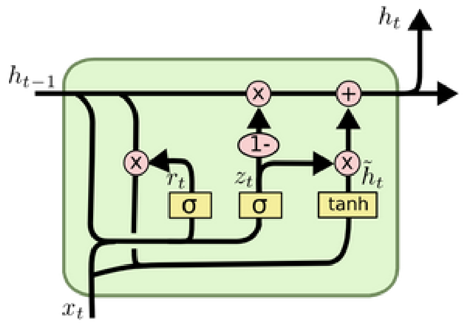
Convolutional Neural Networks (CNNs) are well-suited to many tasks in functional genomics and medical image analysis



- Learn hierarchical feature sets across multiple convolution layers
- Spatially invariant – cannot keep track of the relative order of the input datapoints
- Fixed size inputs
- Applications:
  - Variant prioritization
  - Motif discovery
  - Tumor/lesion detection in medical images
- Residual networks are effective tools for constructing very deep convolutional networks.

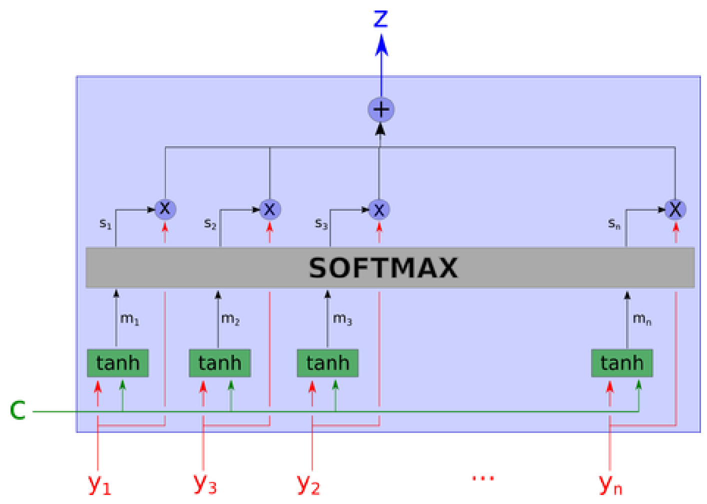
Recurrent neural networks are well-suited to EHR data, text data where sequential information is important

- LSTM's learn long-term dependencies



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

- Attention mechanisms allow for relative weighting of the inputs



Stacked autoencoders are a powerful tool for unsupervised learning

- Clustering and dimensionality reduction
- Learning generative models of data

Denoising autoencoders

- Take a partially corrupted input whilst training to recover the original undistorted input.

Variational autoencoders (VAE)

- Make strong assumptions about the distribution of latent variables.
- Assume data is generated by directed graphical model.

Sparse autoencoders

- Pre-training for classification tasks.

Contractive autoencoders

- Add explicit regularizer in objective function.
- Learn a model that is robust to slight variations of input values.

Conditional random fields

- Account for context
- Image segmentation and object recognition
- Gene finding
- Peptide critical functional region finding
- May be used as a post-processing step to clean up CNN results

Lesson 2: Context is important

Image analysis vs Functional genomics

Image analysis

- 100s - 1000s of layers
  - GoogleNet
- Very small convolution kernels
  - 2x2 or 3x3

Functional genomics

- 1-5 layers
- Larger convolution kernels
  - Basset uses kernel width 11, 9, 7
- Interested in feature representations at multiple layers
  - More emphasis on interpretability

Some general guidelines for learning from sequence data (mileage may vary)

The most common question we got is

“How should I perform hyperparameter search in my model?”

- You guys are definitely asking the right question!
- There is no single answer, but we found this approach works well in many of our models

Some general guidelines for learning from sequence data: Two starting architectures to explore

Bassett model

- 3 convolutional layers with ReLU activations
  - Kernel width of 300,250,250
  - Max pool size/stride of 4,3,3
- 2 fully connected layers with PreLU activations
  - 0.3 Dropout
  - 1000 hidden units in each layer
  - Some amount of L2 regularization
  -

Dragonn base model

- 3 convolutional layers with ReLU activations
  - Kernel width of 15,15,15
  - Single max pool of width/stride 35
- Single fully connected layer

Some general guidelines for learning from sequence data: Suggestions for parameter search

- Try optimizers such as Adam, Adadelta, RMSprop, Adagrad
  - Generally, the default parameter settings for these work well
  - Consider increasing learning rate to speed up training, but watch for any drop in validation / testing performance
- Vary the number of convolution layers from 1 to 5
- Vary the number of convolution filters in each layer from 10 – 400
- Vary kernel width from 5 to 45
- Consider increasing the size of the max pool
- When multiple related labels are available, consider learning from them simultaneously

Lesson 3: Troubleshooting: Common Themes

- **Unbalanced Datasets**

  - Up-sample training examples from the under-represented class of data when performing batch selection
  - Use sample weights to up-weight samples from the under-represented class of data

- **Overfitting to the training data**

  - If possible, add more training examples
  - Smaller kernels – in some application domains
  - Add L1 and L2 regularization
  - Batch normalization
  - Include dropout
  - Make sure you include an 'early-stopping' criterion for training

- **Data is very high-dimensional**

  - Use a variational auto-encoder to encode the data in a lower-dimensional space
  - Train your model on the encoded data

- **Always learn multiple models to account for confidence in prediction.**
    - Rarely done in conventional deep learning
    - Consider K-fold cross-validation
    - Vary parameters (i.e. genomic window sizes)
    - Consider different architectures with similar performance
    - Initialize training with multiple random seeds
    - Sample different negative backgrounds

Future outlook of the field: areas of active research

Model Interpretability

- Identifying important inputs:
  - Perturbation-based approaches (eg: in-silico mutagenesis)
  - Backpropagation-based approaches (gradients, guided backprop, DeepLIFT)

  - Not covered: LIME (Locally Interpretable Model-Agnostic Explanations) - designed for *any* black box - builds a linear model that matches the predictions of the more complex model locally.

- Finding re-occurring patterns:
  - Visualizing maximally activating inputs of individual neurons (gives redundant patterns)
  - MoDISco (Motif Discovery from Importance Scores - consolidates input-level importance scores into re-occurring patterns)

- Identifying dependencies:
  - Delta DeepLIFT focuses on interpreting multi-modal and multi-view models
  - Various approaches in the literature for general black-box models

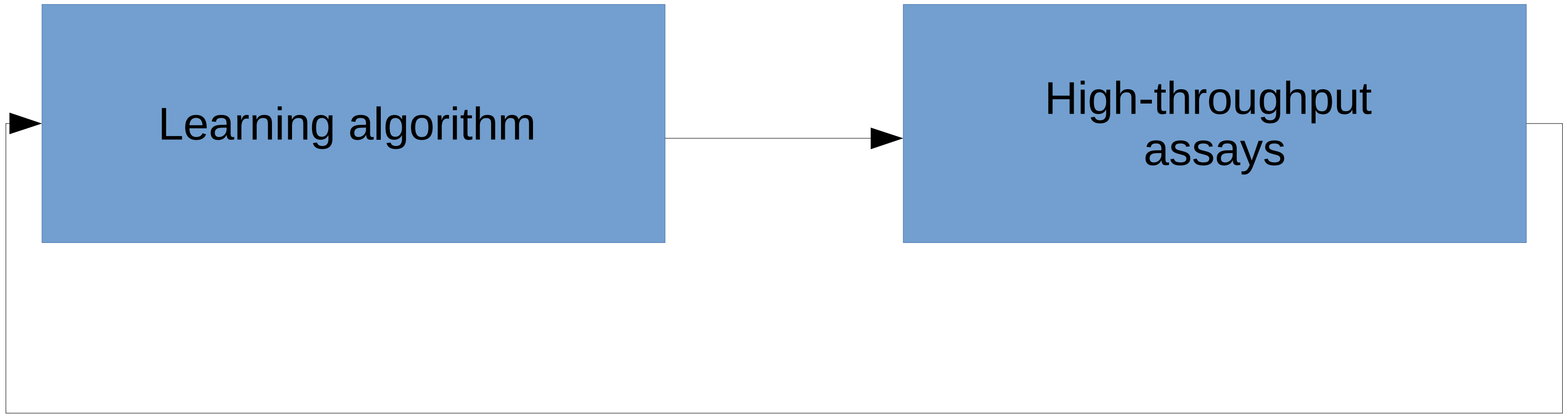
- Learn multiple models to account for confidence in interpretation.
  - Test the effects of jittering inputs
  - Test the effects of different equivalently predictive models (architectures, seeds, window sizes)
  - Test effects of negative background set

Genome-wide models for functional genomics

- Need to efficiently parallelize existing algorithms
- Improvements in data storage/representation
- I/O bottlenecks must be overcome

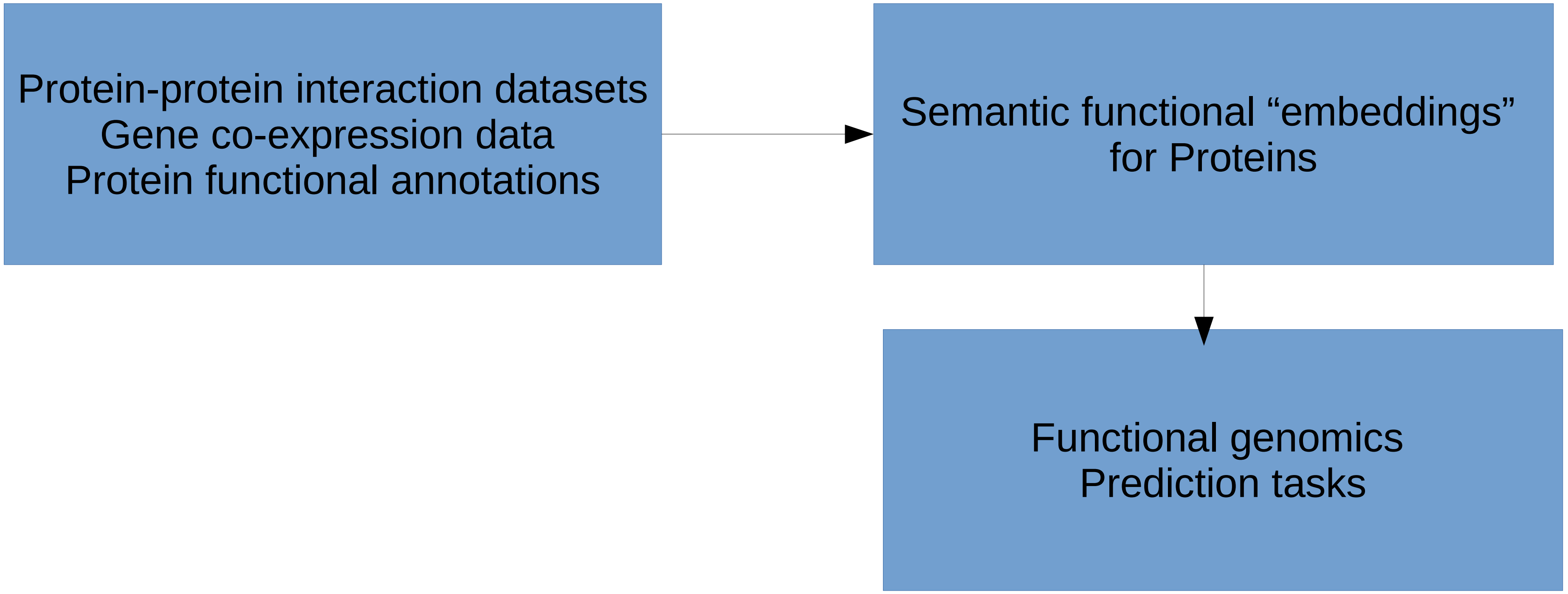
Connecting the loop between automated high-throughput experimental assays (i.e. genome editing screens) and deep learning models

- Can the learning algorithm suggest most useful experiments to perform?
- Perturbation followed by measurement
- Reinforcement learning strategies may help



Transfer Learning: Deep learning has the ability to integrate across domains in a very plug and play manner.

- Biology has many highly connected problems, often with insufficient data
- We can leverage models and data representations from one problem for use in a related problem.
- Pharmacogenomics (protein-drug interactions)
  - Usually, we model explicit properties of proteins (structure) and drugs (chemical composition)


Protein-protein interaction datasets
Gene co-expression data
Protein functional annotations
Semantic functional “embeddings” for Proteins
Functional genomics Prediction tasks

Probabilistic and generative deep architectures: Combining “classical” deep learning with “classical” probabilistic graphical models

- Graphical models have dramatic advantages in terms of interpretability
  - Encode causality
  - Can be used for inference

- PGMs have difficulty inducing novel representations from raw data

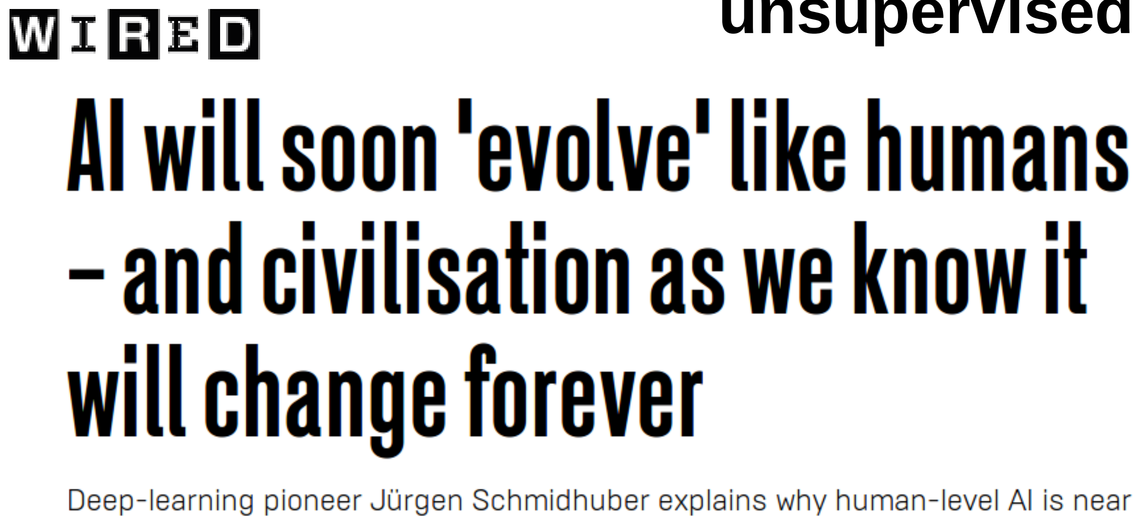
- Solution
  - Use classical deep learning to identify key predictive features in input (i.e. motifs)
  - Use reduced set of predictive features in a PGM to build a more interpretable model

- Long term strategy: fully Bayesian and probabilistic neural networks.

Deep Reinforcement Learning and Semi-Supervised learning

- Deep reinforcement learning leverages reward outcome signals resulting from actions rather than correctly labeled data.
  - Most closely resembles how humans actually learn.
  - Foundational work by Jurgen Schmidhuber : “Evolving large-scale neural networks for vision-based reinforcement learning” (*GECCO,* 2013)
    - video game-based race car driving from high-dimensional visual input streams.
  - DeepMind “Human-level control through deep reinforcement learning” (*Nature,* 2015)
  - **An important step toward distinguishing between correlated and causal variables**
    - Allows the model to take an action to test causation
- Semi-supervised learning exploits both labeled and unlabeled data.
  - Ladder networks add skip connections to CNNS, and simultaneously minimize the sum of supervised and unsupervised cost functions to denoise representations at every level of the model.

Active vs passive unsupervised learning



WIRED

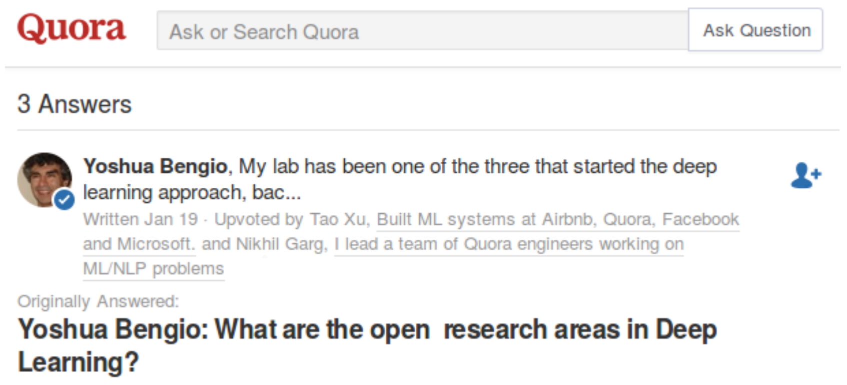
AI will soon 'evolve' like humans – and civilisation as we know it will change forever

Deep-learning pioneer Jürgen Schmidhuber explains why human-level AI is near

22 Nov 2016

- **Passive UL** is simply about detecting regularities in observation streams
  - Learning to encode data with fewer computational resources
  - data compression through predictive coding
- **Active UL** is about learning to shape the observation stream through action sequences that help the learning agent figure out how the world works and what can be done in it.
- *“Active UL explains all kinds of curious and creative behaviour in art and music and science and comedy, and we have already built simple artificial 'scientists' based on approximations thereof. There is no reason why machines cannot be curious and creative.”*

According to Yoshua Bengio...


Quora Ask or Search Quora Ask Question

3 Answers

Yoshua Bengio, My lab has been one of the three that started the deep learning approach, bac...

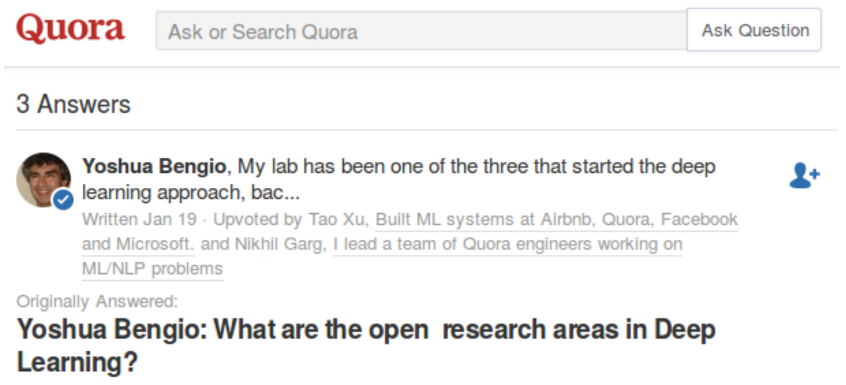
Written Jan 19 · Upvoted by Tao Xu, Built ML systems at Airbnb, Quora, Facebook and Microsoft. and Nikhil Garg, I lead a team of Quora engineers working on ML/NLP problems

Originally Answered:

Yoshua Bengio: What are the open research areas in Deep Learning?

- Unsupervised learning:
  - generative models that generate crisp images and sounds over a wide set of variations covering natural images and sounds
  - semi-supervised learning that makes a difference even when the labeled dataset is not tiny
  - learning a two-way transformation of the data into a space where variables are disentangled (or mostly independent)
  - bringing (iterative) inference back in deep learning to handle non-factorial posteriors over the latent variables

According to Yoshua Bengio...


Quora

3 Answers

Yoshua Bengio, My lab has been one of the three that started the deep learning approach, bac...

Written Jan 19 · Upvoted by Tao Xu, Built ML systems at Airbnb, Quora, Facebook and Microsoft. and Nikhil Garg, I lead a team of Quora engineers working on ML/NLP problems

Originally Answered:

Yoshua Bengio: What are the open research areas in Deep Learning?

- Models of really long-term dependencies in sequential data
  - Think chromosome-level
  - Learners that discover hierarchical representations at multiple time scales
- **Bridging the gap between deep learning and biology**
- **Many applications which are under-explored, and in particular I would like to see much more work in the area of health (with some specific issues like missing values and being to exploit data from small studies via transfer learning)**