Recitation: AutoEncoders, AutoML, Active Learning

Lyle Ungar

Auto-encoders

- **ML mostly uses PCA; medical researchers often use ICA**
  - Why the difference?
- **PCA**
  - Clean math: orthagonality, L2 loss, SVD
  - Fast algorithms: “randomized (thin) SVD”
- **ICA**
  - Potentially nicer explanations

Auto-encoders

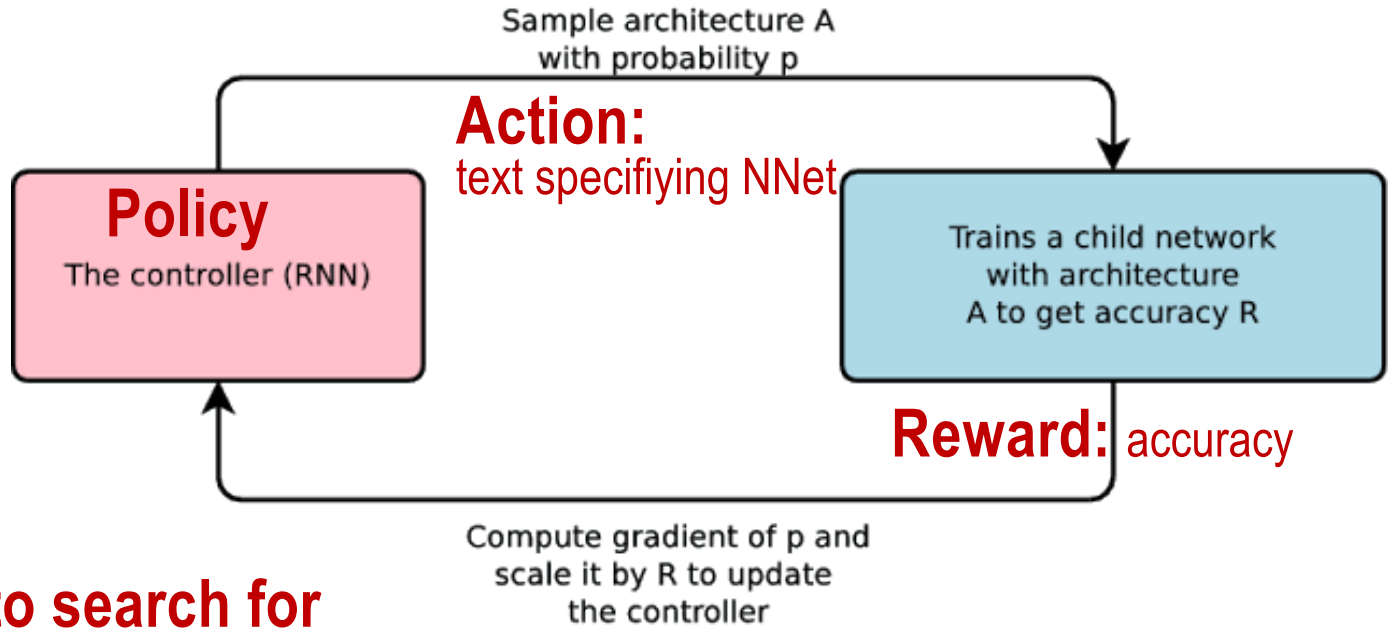
- **What is the manifold learned in PCA?**
- The subspace spanned by the eigenvectors of the covariance matrix that we keep

Auto-ML

- **What is the meta-model learned by auto-sklearn?**
  - **Inputs:** meta-features of a dataset, hyperparameters
  - **Output:** model test accuracy
- **Why is this called "meta-learning" using "meta-features?**
- **What is "Bayesian" about this approach?**

Auto-ML for Deep Learning

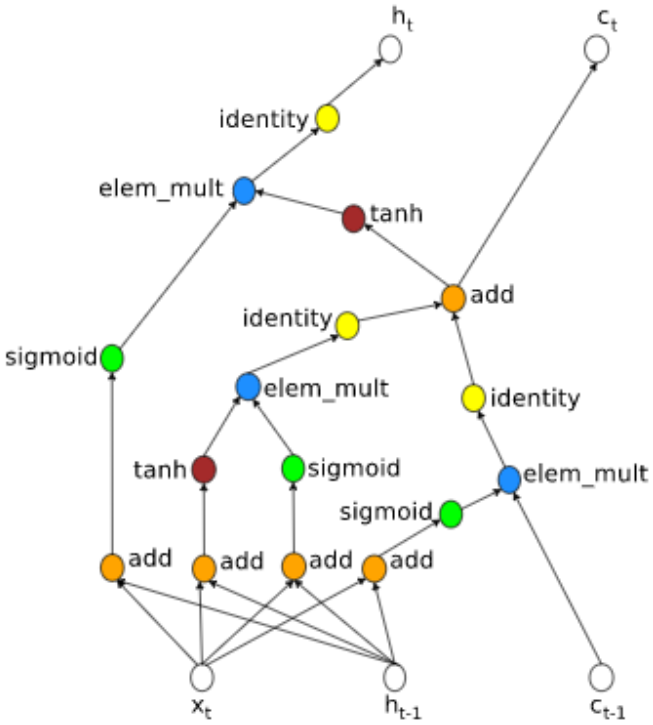
Using Machine Learning to Explore Neural Network Architecture



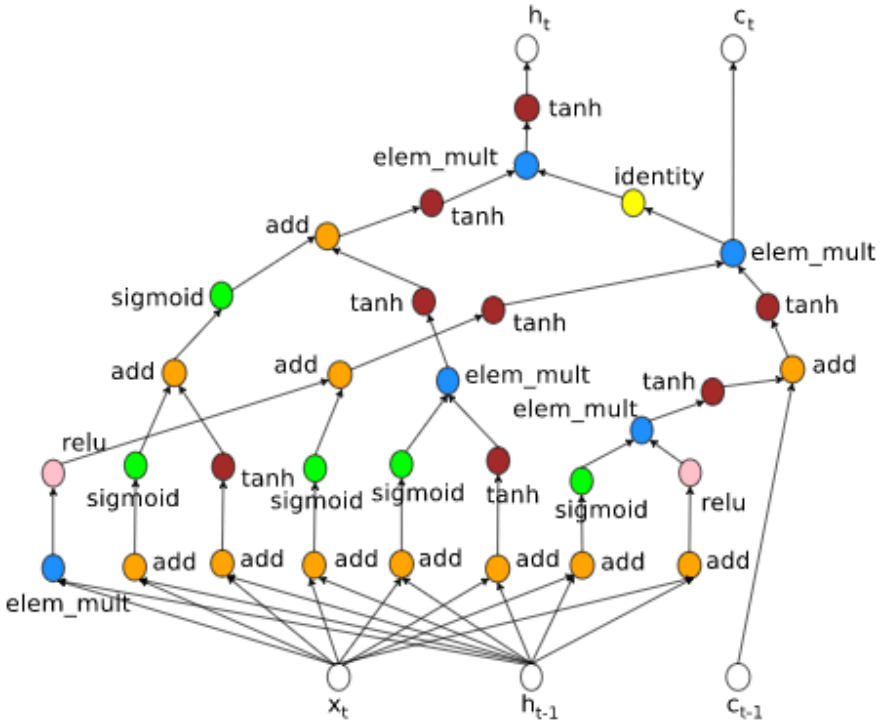
Use RL to search for the 'best' neural net architecture

https://research.googleblog.com/2017/05/using-machine-learning-to-explore.html

AutoML learns network structure



Human built



Learned by RL

https://research.googleblog.com/2017/05/using-machine-learning-to-explore.html

Active Learning

- **Active learning**
  - Uncertainty sampling
  - Query by committee
  - Information-based loss functions
- Optimal experimental design
- Response surface modeling

What is the different between uncertainty sampling and maximizing information gain?

Find A-optimal design for regression

- **Current model**
  - $y = x_1 + 2\,x_2$
- **Current data**
  - **X** = [1 1; 1 2]
- **Which data point is better to label:** (0,0) **or** (2,2)**?**
- **How do you answer this?**

Goal: Minimize variance of *w*

If $y = \mathbf{x}^T\beta + \varepsilon$ **then** $\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$

$\mathbf{w} \sim N(\beta, \sigma^2(\mathbf{X}^T\mathbf{X})^{-1})$ $\varepsilon \sim N(0, \sigma^2)$

We want to minimize the variance of our parameter estimate **w**, so pick training data **X** to minimize $(\mathbf{X}^T\mathbf{X})^{-1}$

But that is a matrix, so we need to reduce it to a scalar

A-optimal (average) design minimizes	**trace** $(X^TX)^{-1}$
D-optimal (determinant) design minimizes	**log det** $(X^TX)^{-1}$
E-optimal (extreme) design minimizes	**max eigenvalue of** $(X^TX)^{-1}$

Alphabet soup of other criteria (C-, G-, L-, V-, etc.)

Find A-optimal design for regression

- **Current data**
  - X = [1 1; 1 2]
- **Which data point is better to label:** (0,0) **or** (2,2)**?**

```
import numpy as np
X = np.array([[1. , 1.],
              [1. , 2.],
              [0. , 0.]])
print('(0,0)', np.trace(np.linalg.inv(X.T@X)))
X = np.array([[1. , 1.],
              [1. , 2.],
              [2. , 2.]])
print('(2,2)',np.trace(np.linalg.inv(X.T@X)))
```

```
(0,0) 7.000000000000006
(2,2) 3.000000000000003
```

Uncertainty sampling

- **Current model**
  - $log(p(y)/(1-p(y)) = x_1 + 2\ x_2$
- **Current data**
  - X = [1 1; 1 2]
- **Which data point is better to label:** (0,0) **or** (2,2)**?**
- **How do you answer this?**

Uncertainty sampling

- **Current model**
  - $log(p(y)/(1-p(y)) = x_1 + 2\ x_2$
- **Which data point is better to label:** (0,0) **or** (2,2)**?**
  - (0,0) : $log(p(y)/(1-p(y)) = 0$
  - (2,2) : $log(p(y)/(1-p(y)) = 6$
- **Which is more uncertain?**
  - (0,0) : $p(y)/(1-p(y)) = 1$
  - (2,2) : $p(y)/(1-p(y)) = e^6$

Response surface modeling

- **Goal:** find $argmin_{\mathbf{x}} f(\mathbf{x})$
- Assume $y = f(x) = w_0 + w_1 x + w_2 x^2$
- **Start with three (x,y) points**
  - *(0,0) (1,-1) (2,1)*
- **What do I do?**

Response surface modeling

- Assume $y = f(x) = w_0 + w_1 x + w_2 x^2$
- **Start with three** *(x,y)* **points**
  - *(0,0) (2,0) (3,3)* -- currently at x=1
- **What do I do?**
  - **Fit model** : $f(x) = 0 - 2x + x^2$
  - **Find better *x***: *x=1*
  - **Observe *y*:** *y = -1*
  - **Repeat**

More questions

- What ways have we seen to create a supervised learning problem from an unsupervised one?
- Why is unsupervised learning so important for deep learning?
- What are the two most important benefits of AutoML?
- …