Conditional Modeling For Fun and Profit

Kyle Kastner

Université de Montréal - MILA
Intern - IBM Watson @ Yorktown Heights
Deep Learning, Simple Concepts

- Universal function approximators
- *Learn* the features
- Desire hierarchy in learned features
  - $y = h(g(f(x))$
  - \{h, g, f\} are nonlinear functions
- Classification
  - Learn $p(y \mid x) = h(g(f(x)))$
Basic Anatomy

- Weights \((W, V)\)
- Biases \((b, c)\)
- Morph features using non-linear functions e.g.
  - \(layer_1_{\text{out}} = \tanh(\text{dot}(X, W) + b)\)
  - \(layer_2_{\text{out}} = \tanh(\text{dot}(layer_1_{\text{out}}, V) + c)\) ...
- Backpropagation to “step” values of \(W, V, b, c\)

[1, 2]
Mixture Density Networks

- What are sufficient statistics?
  - Describe an instance of a distribution
  - Gaussian with mean $u$, variance $s$
  - Bernoulli with probability $p$

- Ties to neural networks
  - Arbitrary output parameters
  - Can we interpret parameters in a layer as sufficient statistics? YES!
  - Cost / regularization forces this relationship

[3, 1]
Parameterizing Distributions

- sigmoid -> Bernoulli
- softmax -> Multinomial
- linear, linear -> Gaussian with mean, log_var
- softmax, linear, linear, linear -> Gaussian mixture
- Can combine with recurrence
  - Learned, dynamic distributions over sequences
  - *Incredibly* powerful

[3, 1, 4, 5, 6, 7, 8, 9]
Latent Factor Generative Models

- Auto-Encoding Variational Bayes
  D. Kingma and M. Welling
    - Model known as Variational Autoencoder (VAE)
    - See also Stochastic Backpropagation and Approximate Inference in Deep Generative Models, Rezende, Mohamed, Wierstra

[11, 12, 13]
ENCODER  [11, 12, 13]  DECODER
A Bit About VAE

- Want to do latent variable modeling
- Don’t want to do MCMC or EM
- Sampling Z blocks gradient
- Reparameterization trick
  - Exact soln intractable for complex transforms (like NN)
  - Lower bound on likelihood with KL divergence
  - N(mu, sigma) -> mu + sigma \* N(0, 1)
  - Like mixture density networks, but in the middle
  - Now trainable by backprop

[11, 12, 13]
Taking The Wheel

- **Specifics of MNIST digits**
  - Writing style and class
  - Traits are semi-independent
  - Can encode this in the model
  - $y \rightarrow$ softmax classifier ($\sim y$ is sample)
  - $p(z | x, y), p(z | x, \sim y) \text{ or } p(z | x, f(x))$

- **Fully conditional version of M2**
  - Semi-Supervised Learning with Deep Generative Models, Kingma, Rezende, Mohamed, Welling [13, 14]
Conditioning, Visually

[13, 14]
In Practice...

- Conditioning is a strong signal
  - $p(x_{\text{hat}} \mid z)$ vs. $p(x_{\text{hat}} \mid z, y)$
- Can give control or add prior knowledge
- Classification is an even stronger form
  - Prediction is learned by maximizing $p(y \mid x)$!
  - In classification, don’t worry about forming a useful $z$

[1, 13, 14]
Conditioning Feedforward

- Concatenate features
  - `concatenate(((X_train, conditioning), axis=1))`
  - \( p(y \mid X_1 \ldots X_n, L_1 \ldots L_n) \)
- One hot label \( L \) (scikit-learn `label_binarize`)
- Could also be real valued
- Concat followed with multiple layers to “mix”

[1]
Convolution and Recurrence

- Exploit structure and prior knowledge
  - Parameter sharing is strong regularization
- Convolution - exploit locality
  - $p(y | X_{i-n} \ldots X_{i+n}) \times p(y | X_{i+1-n} \ldots X_{i+1+n}) \ldots$
  - A learned filter over a fixed 1D or 2D window
  - Window slides over all input, updates filter
- Recurrence - exploit sequential information
  - $p(y | X_1 \ldots X_t) = p(y | X_{<=t})$ can be seen as:
  - $\sim p(y | X_1) \times p(y | X_2, X_1) \times p(y | X_3, X_2, X_1) \ldots$

[1, 4, 5, 6, 7, 8, 9]
More on Recurrence

- Hidden state \( (s_t) \) encodes sequence info
  - \( p(X_{\leq t}) \) (in \( s_t \)) is *compressed representation* of \( X \)

- Recurrence similar to
  - Hidden Markov Model (HMM)
  - Kalman Filter (KF, EKF, UKF)

[1, 4, 15, 16]
How-To MDN + RNN

- Generating Sequences with Recurrent Neural Networks
  Alex Graves

- Multi-level RNN, outputs GMM and bernoulli
  - Handwriting
    - Pen up/down and relative position per timestep
  - Vocoder representation of speech
    - Voiced/unvoiced and MFCC per timestep [3, 4]
How-To Continued

- **Conditional model**
  - Adds input attention (more on this later)
  - Gaussian per timestep over one hot text
  - \( p(\text{bernoulli, GMM} \mid X_t, \text{previous state, focused text}) \)
  - This gives *control* of the output via input text

http://www.cs.toronto.edu/~graves/handwriting.html  https://www.youtube.com/watch?v=-yX1SYeDHbg&t=43m30s

[3, 4]
Similar Approaches

- RNN with sigmoid output
  - ALICE
- RNN with softmax
  - RNN-LM
- RNN-RBM, RNN-NADE

[3, 1, 4, 5, 6, 7, 8, 9]
Research Questions

● Possible Issues
  ○ Prosody/style are not smooth over time
  ○ Deep network, but still shallow latent variables
  ○ Vocoder is a highly engineered representation

● How can we fix these problems?
  ○ First, a bit about conditioning in RNNs
Conditioning In Recurrent Networks

- RNNs model $p(X_t | X_{<t})$
- Initial hidden state can condition
  - $p(X_t | X_{<t}, c)$ where $c$ is init. hidden state (context)
- Condition by concatenating in feedforward
  - Before recurrence or after
- Can do *all of the above*
Conditioning with a Sequence

- RNN outputting Gaussian parameters over seq
  - Seen in Generating Sequences

- Use an RNN to compress
  - Hidden state encodes $p(X_{<=t})$
  - Project into init hidden and ff
  - Now have $p(y_t | y_{<t}, X_{<=t})$
  - Known as RNN Encode-Decode
  - Cho et al

[16, 17]
Distributing The Representation

- Distribute context, Bahdanau et al
- Bidirectional RNN
  - $p(X_i \mid X_{<i}, X_{>i})$ for $i$ in $t$
  - Needs whole sequence
  - But sometimes this is fine
- Soft attention over hiddens
- Choose what is important

[16, 17, 18]
Previously, on FOX...

- **RNN-GMM Issues**
  - Prosody/style are not smooth over time
  - Deep network, but still shallow latent variables
  - Vocoder is a highly engineered representation

- **How can we try to fix these problems?**
  - Distributed latent representation for Z
  - Use modified VAE to make latents deep
  - Work on raw timeseries inputs
    - Extreme approach, but proves a point
Existing Approaches

- VRAE, $Z_t$ independent
- STORN, $Z_t$ independent
- DRAW, $Z_t$ loosely dependent via canvas
- No large scale real-valued experiments
  - VRAE, no real valued experiment
  - STORN, real valued experiment was small
  - DRAW, real values weren’t sequences

[18, 19, 20]
Variational RNN

- **Speech**
  - Complex but structured noise driven by mechanics
  - Ideal latent factors include these mechanics
- **$Z_{<t}$ should affect $Z_t$ and $h_t$**
- **Use a recurrent prior**
Primary Functions

\[ p(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T}) = \prod_{t=1}^{T} p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{x}_{< t}) p(\mathbf{z}_t \mid \mathbf{x}_{< t}, \mathbf{z}_{< t}). \]

\[ \mathbf{z}_t \mid \mathbf{x}_t \sim \mathcal{N}(\mu_{z,t}, \text{diag}(\sigma_{z,t}^2)) , \text{ where } [\mu_{z,t}, \sigma_{z,t}] = \varphi_T^{\text{enc}}(\varphi_T^{\mathbf{x}}(\mathbf{x}_t), \mathbf{h}_{t-1}) \]

\[ \mathbf{x}_t \mid \mathbf{z}_t \sim \mathcal{N}(\mu_{x,t}, \text{diag}(\sigma_{x,t}^2)) , \text{ where } [\mu_{x,t}, \sigma_{x,t}] = \varphi_T^{\text{dec}}(\varphi_T^{\mathbf{z}}(\mathbf{z}_t), \mathbf{h}_{t-1}) \]

\[ \mathbf{z}_t \sim \mathcal{N}(\mu_{0,t}, \text{diag}(\sigma_{0,t}^2)) , \text{ where } [\mu_{0,t}, \sigma_{0,t}] = \varphi_T^{\text{prior}}(\mathbf{h}_{t-1}) \]

\[ \mathbf{h}_t = f_\theta(\varphi_T^{\mathbf{x}}(\mathbf{x}_t), \varphi_T^{\mathbf{z}}(\mathbf{z}_t), \mathbf{h}_{t-1}) \]

\[ + \sum_{t=1}^{T} -\text{KL}(q(\mathbf{z}_t \mid \mathbf{x}_{\leq t}, \mathbf{z}_{< t}) \| p(\mathbf{z}_t \mid \mathbf{x}_{< t}, \mathbf{z}_{< t})) \]

\[ + \mathbb{E}_{q(\mathbf{z}_t \mid \mathbf{x}_{\leq t}, \mathbf{z}_{< t})} [\log(p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{z}_{< t}))]. \]

[15]
Prior

- Used for KL divergence
- Fixed in VAE to $N(0, 1)$
- Here it is learned
- Instead of “be simple” (as in VAE), this says “be consistent”

\[
\sum_{t=1}^{T} -\text{KL}(q(z_t | x_{\leq t}, z_{< t}) \| p(z_t | x_{< t}, z_{< t}))
\]
\[
+ \mathbb{E}_{q(z_t | x_{\leq t}, z_{< t})} [\log(p(x_t | z_{\leq t}, x_{< t}))].
\]

\[
z_t \sim \mathcal{N}(\mu_{0,t}, \text{diag}(\sigma^2_{0,t})) , \text{ where } [\mu_{0,t}, \sigma_{0,t}] = \varphi_T^{\text{prior}}(h_{t-1})
\]
Inference (encode)

- Previous hidden state
  - $h_{t-1}$
- Data
  - $X_t$
- Hidden state information
  - $z_{<t}$
  - $X_{<t}$

$$h_t = f_\theta (\varphi^x_T(x_t), \varphi^z_T(z_t), h_{t-1})$$

$$z_t \mid x_t \sim \mathcal{N}(\mu_{z,t}, \text{diag}(\sigma^2_{z,t}))$$, where $[\mu_{z,t}, \sigma_{z,t}] = \varphi^{\text{enc}}_T(\varphi^x_T(x_t), h_{t-1})$
Generation (decode)

- Generate based on
  - $Z_t, h_{t-1}$
  - $h_{t-1}$ has $z_{<t}, X_{<t}$
  - $Z_t$ has $z_{<t}, X_{<=t}$

$$h_t = f_{\theta} (\varphi_x^z(x_t), \varphi_z^z(z_t), h_{t-1})$$

$$x_t \mid z_t \sim \mathcal{N} (\mu_{x,t}, \text{diag}(\sigma_{x,t}^2)) \text{, where } [\mu_{x,t}, \sigma_{x,t}] = \varphi_{\tau}^{\text{dec}} (\varphi_{\tau}^z(z_t), h_{t-1})$$

[15]
Recurrence

- Just a regular RNN
- Input projection is a VAE
- Can use LSTM, GRU, others

\[
\mathbf{h}_t = f_\theta (\varphi^x_T(\mathbf{x}_t), \varphi^z_T(\mathbf{z}_t), \mathbf{h}_{t-1})
\]

\[
p(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T}) = \prod_{t=1}^{T} p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{x}_{< t}) p(\mathbf{z}_t \mid \mathbf{x}_{< t}, \mathbf{z}_{< t}).
\]
KL Divergence
Learned Filters

(a) $\varphi_T^{\text{enc}}$

(b) $\varphi_T^{\text{dec}}$
Final Thoughts on VRNN

- Empirically, structured Z seems to help
  - Keep style consistent
  - Predict very correlated data, like raw timeseries
  - Also works well for unconditional handwriting

RNN-GMM  VRNN-GMM
Takeaways and Opinions

- Can use deep learning like graphical modeling
  - Different tools, same conceptual idea
  - Conditional probability modeling is key
- Put knowledge in model structure, not features
- Let features be learned from data
- Use conditioning to control or constrain
Thanks!

@kastnerkyle

Slides will be uploaded to https://speakerdeck.com/kastnerkyle
References (1)


References (2)


More on Convolution

- Define size of feature map and how many
  - Similar to output size of feedforward layer

- Parameter sharing
  - Small filter moves over entire input
  - Believe local statistics consistent over regions
  - Enforced by parameter sharing

- Condition by concatenating
  - Along “channel” axis
  - [Link](http://arxiv.org/abs/1406.2283)