Deep Learning

~~Tips from the Road~~

CRASH COURSE

Kyle Kastner

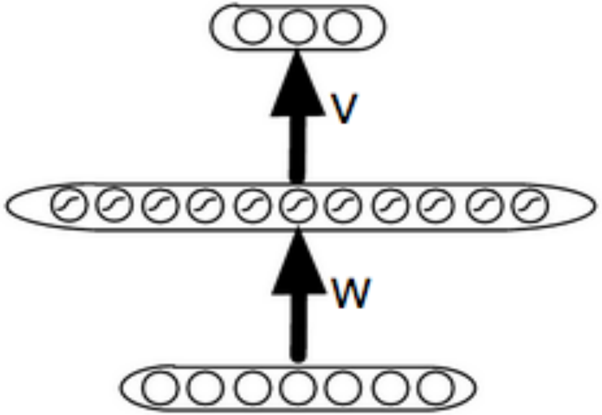
Université de Montréal - MILA

Intern - IBM Watson @ Yorktown Heights

Automation Spectrum

Introspection Statistics | Machine Learning | Automation Deep Learning

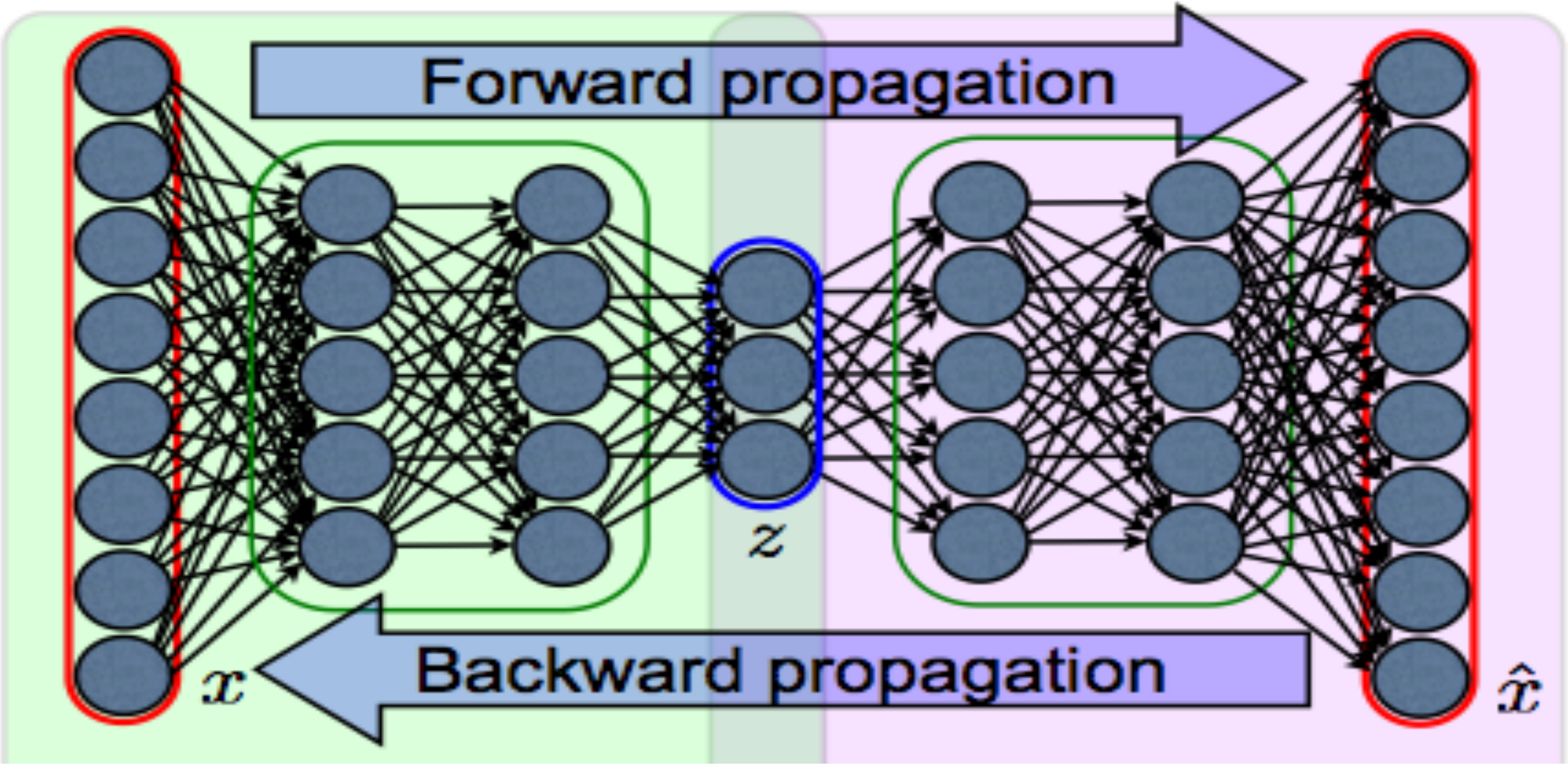
Basic Anatomy



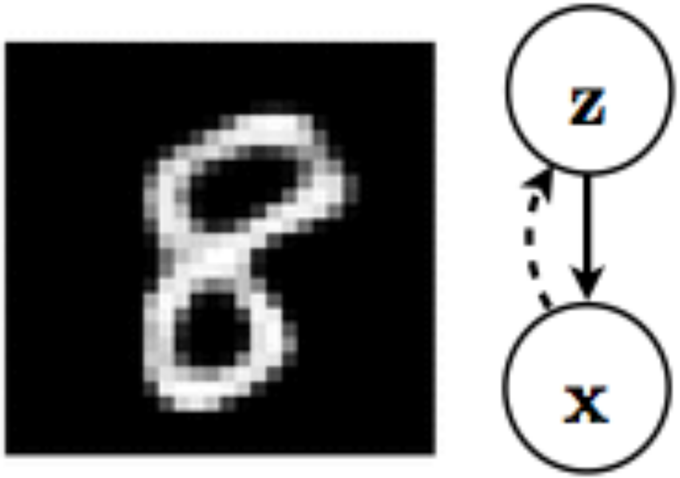
- Weights (**W, V**)
- Biases (**b, c**)
- Init weights randomly, biases can start at 0
- Morph features using non-linear functions
  - layer_1_out = tanh(dot(W, X) + b)
  - layer_2_out = tanh(dot(V, layer_1_out) + c) ...
- Backpropagation to “step” values of **W**,**V**,**b**,**c**

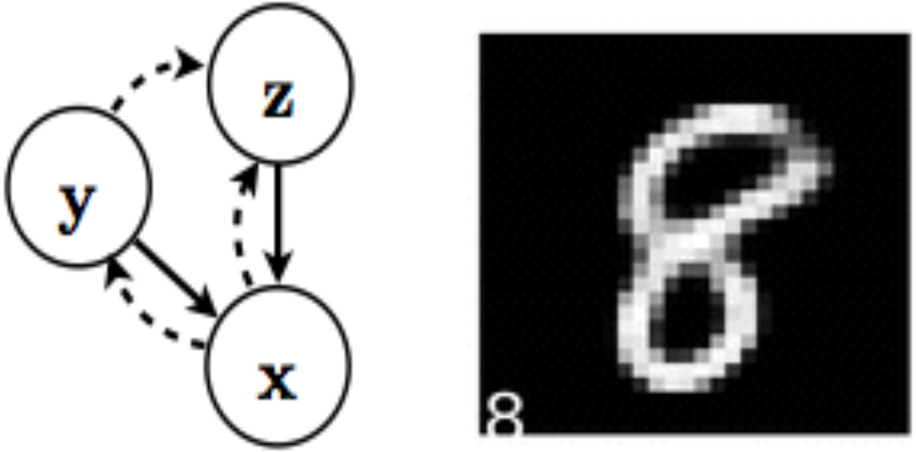
General Guidelines

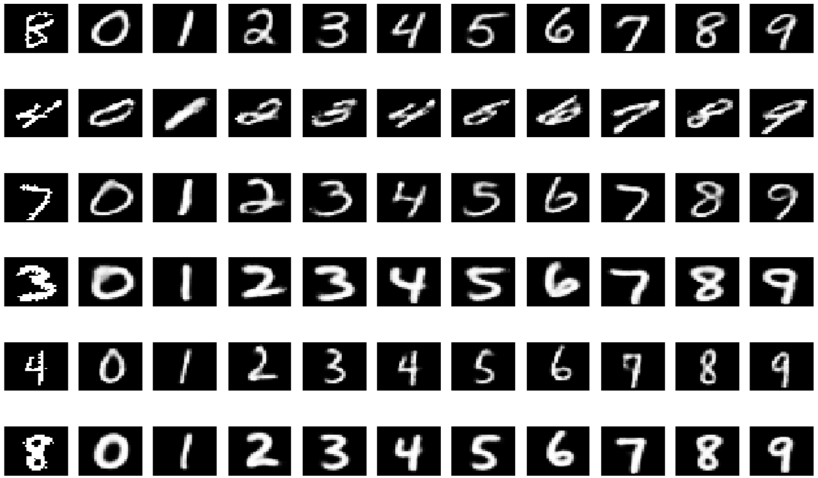
- Prefer rectified activation (ReLU)
    - `def relu(X): return X * (X > 0)`
- Optimization
    - RMSProp w. momentum, ADaM (easiest to tune)
    - Stochastic Gradient Descent w. momentum (harder)
- Regularize with Dropout
    - https://www.cs.toronto.edu/~**hinton**/csc2535/notes/lec6a.**ppt**
- Great initialization reference
    - https://plus.google.com/+SoumithChintala/posts/RZfdrRQWL6u



ENCODE DECODE







Conditioning, Visually

In Practice...

- Conditioning is a *strong* signal
  - p(x_hat | z) vs. p(x_hat | z, y)
- Can give *control* or add prior knowledge
- Classification is an even stronger form
  - Prediction is learned by maximizing p(y | x) !
  - In classification, don’t worry about forming a useful z

Conditioning Feedforward

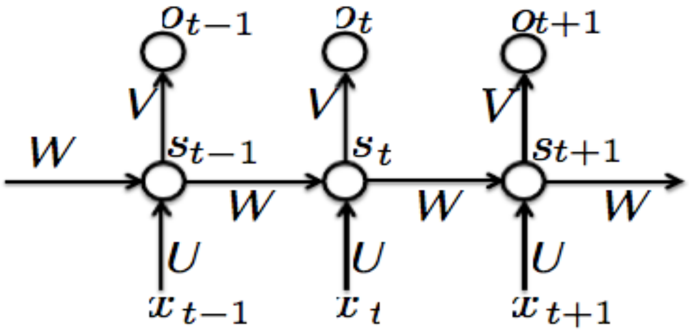
- Concatenate features
  - concatenate((X_train, conditioning), axis=1)
  - p(y | X_1 … X_n, L_1 … L_n)
- One hot label L (scikit-learn label_binarize)
- Could also be real valued
- Concat followed with multiple layers to “mix”

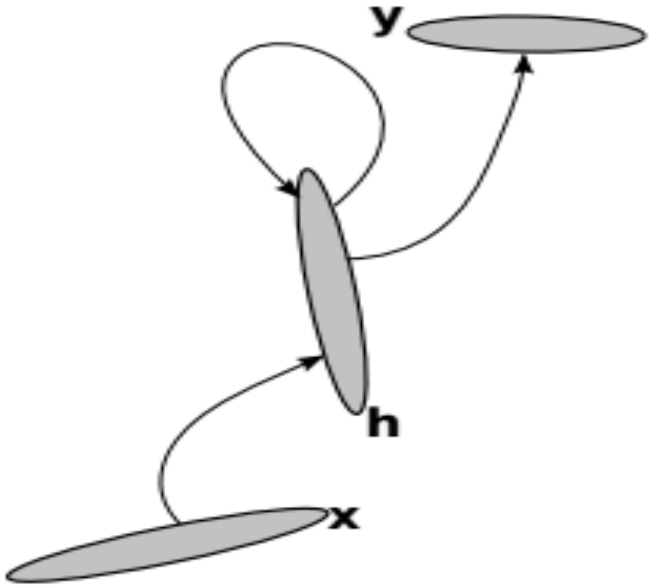
Convolution and Recurrence

- Exploit structure and prior knowledge
  - Locality, neighbors have key information (convolution)
  - Sequence, ordering is crucial (recurrence)
- Convolution (Not discussed here)
- Recurrence
  - $p(y \mid X_1 \ldots X_n)$ can be seen as:
  - $\sim p(y \mid X_1) * p(y \mid X_2, X_1) * p(y \mid X_3, X_2, X_1) \ldots$

More on Recurrence

- Hidden state (s_t) encodes sequence info
  - X_1 … X_t, but *compressed*
- Recurrence similar to
  - Hidden Markov Model (HMM)
  - Kalman Filter

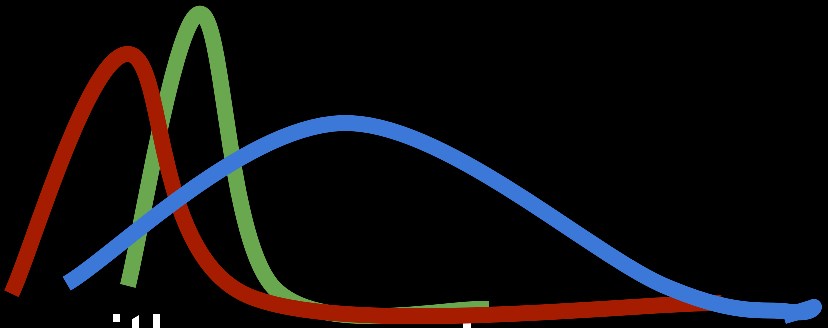




Yet More On Recurrence

- Initialize hidden state to be orthogonal
  - Use U from U, S, V = svd(randn_init)
- Long short term memory or gated recurrent
  - {LSTM, GRU} fancy recurrent activations
- {Sentences, dialogues, sounds} are sequences
  - Many-to-one (sequence recognition)
  - Many-to-many (sequence to sequence)
  - One-to-many (sequence generation)
  - Many-to-one-to-many (encode-decode)

Parameterizing Distributions



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

Where is it used?

- Image classification (conv)
- Text-to-text translation (rec encode-decode)
- Q&A systems / chatbots (rec encode-decode)
- Speech recognition (rec or conv)
- Speech synthesis (rec with GMM output)

Future Directions

A giraffe standing in a forest with trees in the background.

- Attention
- Dedicated memory
  - Separate “what” from “where”
  - Similar to attention
- Combine with reinforcement learning
  - No more labels?
  - Deep Q Learning - playing Atari from video!
  - https://www.youtube.com/watch?v=V1eYniJ0Rnk#t=1m12s

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

@kastnerkyle

Thanks!

Repo with slides and links https://github.com/kastnerkyle/SciPy2015
Slides will be uploaded to https://speakerdeck.com/kastnerkyle

sklearn-theano, a scikit-learn compatible library for using pretrained networks : http://sklearn-theano.github.io/

Neural network tutorial by @NewMu / Alec Radford : https://github.com/Newmu/Theano-Tutorials

Theano Deep Learning Tutorials: http://deeplearning.net/tutorial/

References and Links

Deep Learning Book (Goodfellow, Courville, Bengio): http://www.iro.umontreal.ca/~bengioy/dlbook/

Deep Learning Course (Courville): https://ift6266h15.wordpress.com/

Deep Learning Course (Larochelle): https://www.youtube.com/playlist?list=PL6Xpj9I5qXYEcOhn7TqghAJ6NAPrNmUBH

Encode Decode with Attention (Bahdanau, Cho, Bengio): http://arxiv.org/abs/1409.0473

Caption Generation (Xu, Ba, Kiros et. al.): http://arxiv.org/abs/1502.03044

Generating Sequences with Recurrent Neural Networks (Graves): http://arxiv.org/abs/1308.0850

Depth Map Prediction From A Single Image (Eigen, Puhrsch, Fergus): http://arxiv.org/abs/1406.2283

Semi Supervised Learning (Kingma, Rezende, Mohamed, Welling): http://arxiv.org/abs/1406.5298

Neural Networks Coursera (Hinton): https://www.coursera.org/course/neuralnets

Understanding The Difficulties in Training Deep Feedforward Networks (Glorot, Bengio): http://jmlr.org/proceedings/papers/v9/glorot10a/glorot10a.pdf

Advances in Optimizing Recurrent Nerural Networks (Pascanu, Boulanger-Lewandowski, Bengio): http://arxiv.org/abs/1212.0901

CUT SLIDES

Where is it used?

- Image classification
- Text-to-text translation
- Q&A systems / chatbots
- Speech recognition
- Speech synthesis
- **Usually many, many datapoints**

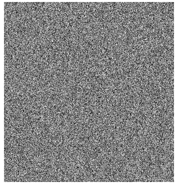
Deep Learning, Simple Concepts

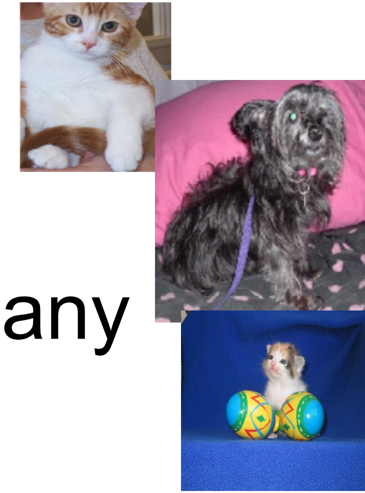
- Universal function approximators
- *Learn* the features
- Expect hierarchy in learned features
  - $y = h(g(f(x)))$
  - $\{h, g, f\}$ are functions
- Classification
  - Learn $p(y \mid x) = h(g(f(x)))$

When To Use Feedforward

- Use for modeling p(y | X_1 … X_n)
  - X_1 … X_n represent features
- Initialize with Xavier method
  - +- 4 * sqrt(6. / (in_sz + out_sz)) uniform
  - +- 0.01 uniform or 0.01 std randn can work
  - Great reference: https://plus.google.com/+SoumithChintala/posts/RZfdrRQWL6u

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
    - Local statistics consistent over all regions
- Condition by concatenating
    - Along “channel” axis
    - http://arxiv.org/abs/1406.2283