# Deep Learning + HEP

Warning: half-baked slides

# Today's talk

Brief overview of deep learning

- Overview of the field
- Where is deep learning right now?
- Spatial data + CNNs
- Temporal data + RNNs
- Training + tricks:
    - regularization and normalization

**Project ideas:** How can it be applied to HEP?

- Deep learning challenges for tracking / particle physics
- Learning probabilistic spatiotemporal dynamics with deep learning
- Generative modeling

# References

- www.deeplearningbook.org -- good book
- http://cs231n.github.io/ -- Deep learning + Computer Vision
- https://cs224d.stanford.edu/ -- Deep learning + Natural Language Processing
- https://www.cs.toronto.edu/~hinton/csc2535/notes/lec10new.pdf -- Hinton’s RNN lecture

# References -- Implementations

- Tensorflow -- www.tensorflow.com
- Theano -- http://deeplearning.net/software/theano/
- Torch -- http://torch.ch/

**Top-level wrappers** → start here

- Keras (TF + Theano) -- www.keras.io
- TFLearn (TF) -- http://tflearn.org/
- Lasagne (Theano) -- https://lasagne.readthedocs.io/en/latest/

# Brief overview of ML / DL

- Popular model class
  - Outperforms other model classes (kernel methods, etc.) in many applications
  - Not always the best (random forests…)
  - Jungle of papers with hacks and tricks → many things do **not** consistently work
- Implement “highly non-convex” $F : X \rightarrow Y$
- Empirical science ~ little to no theory
  - Bag of tricks

# Brief overview of ML / DL

- Supervised learning ~ discriminative $P(y|x)$
- Unsupervised learning ~ generative models $P(x)$
- Semi-supervised

# Brief overview of ML / DL

- Supervised learning ~ discriminative models $P(y|x)$
  - Image recognition → CNNs
  - Sequential prediction → RNNs

# Brief overview of ML / DL

- Unsupervised learning ~ generative models P(x)
  - Auto-encoders -- deterministic P(z|x) -- train through back-prop
  - Variational methods -- stochastic z ~ P(z)
  - Generative Adversarial Networks

# How do you train them?

- A neural net implements F: $X \rightarrow Y$
  - X: images, sentences, ...
  - Y: discrete classes, continuous ...
  - F: (conditional) probability,
- Training objective
  - Multi-class classification: $P(y|x)$ -- x-ent minimization
  - Regression: L2-minimization
- **In practice:**
  - **first overfit, then regularize**
  - **"Bigger / deeper is better"**

# How do you train them?

- Regularization
  - Dropout
  - L1 / L2 / ...
- Normalization
  - Batch-normalization
    - We want to train on a statioinary distribution P(“input”) → standardize activations!
  - Layer-normalization

# Convolutional neural networks

# Tasks

- Classification
- Segmentation
- Structured prediction

# Basics

- Kernel, stride
- Pooling (max, average, …)
- Neural representations: activation statistics encode predictive patterns in the input
- Neural style: activation correlation statistics encode style

# Some state-of-the-art conv-nets

Benchmark datasets

- ImageNet
- MSCoco

Variations

- Inception
- ResNet

# Recurrent neural networks

# Basics

- Update equations
- Non-Markovian dynamics
- Hard to train on long sequences
  - Vanishing / exploding gradients

# Memory in RNNs

- http://colah.github.io/posts/2015-08-Understanding-LSTMs/
    - LSTM
    - GRU
- Non-Markovian dynamics
- Still hard to train on long sequences

# More RNNs

- Highway networks

# RNN-based models

- NLP = learn over **flat** text
  - Large vocabulary → semi-supervised / unsupervised training
- RNNs often are **not** that much better than Bag-of-Words models
  - Need lots of labeled data if trained from scratch
- Unsupervised training over large corpus of **unlabeled** text
  - Word vectors ~ continuous bag-of-words
    - GloVe: model co-occurrence statistics
  - Paragraph vectors
  - Language model: given word w_{t-1}, what's w_t?

# RNN-based models

- Trading space for time
  - Adaptive computation time
- Variable input → Fixed representation
  - Question answering with RNNs
  - Fixed-length span representations
  - **Seems to not be that important, attention is much more powerful**
- Bidirectional RNNs

# RNN-based models

- Prediction
  - Classification: sentiment
  - Structure: Parse-trees
- Sequence-to-sequence models
  - Translation
  - Model conditional probability P(w_t | w_t_1)
  - Scheduled sampling
    - W_t-1 = previous prediction
    - W_t-1 = ground truth
- **Sequence-to-sequence models + attention**

# RNN-based models

- Other lessons
  - Data augmentation works... a bit
    - Label smearing
  - Multi-hop (trade space for time, very hard to train) is hard

# State-of-the-art

- Unsupervised pre-training: boosts perf on problems over long input sequences
    - Pre-train as language model
    - Match statistics!
    - calibrate hidden-hidden weights to propagate gradients over long-distance
- Batch / layer-normalization
- Dropout
    - Be careful not to stop gradients to get to distance past

# ML + sports

# Learning / tracking in sports

**Imitation learning using tracking data**

- Ball prediction
- Player movement prediction
- Hierarchical policies

**Lessons**

- Sparse input → use multi-resolution
- Hierarchy is hard to train
- Multi-task to separate timescales

# Unsupervised learning

# Unsupervised learning

- Auto-encoders with sliding windows
- GAN -- https://arxiv.org/pdf/1406.2661v1.pdf
  - Hard to stabilize
  - SeqGAN -- unproven
- Vision
  - PixelCNN
  - PixelRNN

# Deep learning $\rightarrow$ particle physics

# Project ideas

- Can we incorporate discrete latent variables to model particle decay processes?
- Can we generate tracks that match data distribution?
- Address distributional shift → big problem in RL, robotics, ...
- Structured, hierarchical policies

# Deep learning → particle physics

# Deep learning for HEP tracking

- $F: X \rightarrow Y$
  - X: what does our input look like?
  - Y: what do we want to predict?
- Basketball: player / ball prediction examples
- Challenges
  - Scale
  - Sparsity of input
  - Avoid distributional shift
- What data do we need?
  - Volume
  - Format