CMPT 733 – Big Data Programming II

Deep Learning I

Instructor
Steven Bergner

Course website
https://coursys.sfu.ca/2024sp-cmpt-733-g1/pages
Overview

Renaissance of artificial neural networks
  • Representation learning vs feature engineering

Background
  • Linear Algebra, Optimization
  • Regularization
  • Construction and training of layered models

Frameworks for deep learning
Representations matter

- Transform into the right representation
- Classify points simply by threshold on radius axis
- Single neuron with non-linearity can do this

[Goodfellow, Bengio, Courville 2016]
Depth: Layered composition

[Goodfellow, Bengio, Courville 2016]
Computational graph

[Goodfellow, Bengio, Courville 2016]
Components of learning

- Hand designed program
  - Input $\rightarrow$ Output
- Increasingly automated
  - Simple features
  - Abstract features
  - Mapping from features

[Goodfellow, Bengio, Courville 2016]
Growing dataset size

MNIST dataset

[Goodfellow, Bengio, Courville 2016]

Steven Bergner - CMPT 733
Basics

Linear Algebra and Optimization
Linear algebra

- Tensor is an array of numbers
  - Multi-dim: 0d scalar, 1d vector, 2d matrix/image, 3d RGB image
- Matrix (dot) product \( C = AB \)
  \[ C_{i,j} = \sum_k A_{i,k} B_{k,j} \]
- Dot product of vectors A and B
  - \((m = p = 1 \text{ in above notation})\)

[Goodfellow, Bengio, Courville 2016]
Linear algebra: Norms

- $L^p$ norm

$$||\mathbf{x}||_p = \left( \sum_i |x_i|^p \right)^{\frac{1}{p}}$$

- Most popular norm: L2 norm, $p=2$

- L1 norm, $p=1$: $||\mathbf{x}||_1 = \sum_i |x_i|$.

- Max norm, infinite $p$: $||\mathbf{x}||_\infty = \max_i |x_i|$.

[Goodfellow, Bengio, Courville 2016]
Nonlinearities

- ReLU
- Softplus
- Logistic Sigmoid

[Goodfellow, Bengio, Courville 2016]
Approximate optimization

Ideally, we would like to arrive at the global minimum, but this might not be possible.

This local minimum performs nearly as well as the global one, so it is an acceptable halting point.

This local minimum performs poorly and should be avoided.
Gradient descent

For $x < 0$, we have $f'(x) < 0$, so we can decrease $f$ by moving rightward.

For $x > 0$, we have $f'(x) > 0$, so we can decrease $f$ by moving leftward.

Global minimum at $x = 0$. Since $f'(x) = 0$, gradient descent halts here.

$\begin{align*}
  f(x) &= \frac{1}{2}x^2 \\
  f'(x) &= x
\end{align*}$
Critical points

Saddle point – 1st and 2nd derivative vanish

Poor conditioning:
1. deriv large in one and small in another direction

[Goodfellow, Bengio, Courville 2016]
Optimization algorithm

- Lots of variants address choice of learning rate
- See Visualization of Algorithms
- AdaDelta and RMSprop often work well
Neural network playgrounds

- [http://playground.tensorflow.org/](http://playground.tensorflow.org/)
  - Try out simple network configurations on TF Playground

- [https://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html](https://cs.stanford.edu/people/karpathy/convnetjs/demo/classify2d.html)
  - Visualize linear and non-linear mappings
Regularization

Reduced generalization error without impacting training error
Constrained optimization

- Squared L2 encourages small weights
- L1 encourages sparsity of model parameters (weights)

[Goodfellow, Bengio, Courville 2016]
Dataset augmentation

- Affine Distortion
- Noise
- Elastic Deformation
- Horizontal flip
- Random Translation
- Hue Shift

[Goodfellow, Bengio, Courville 2016]
Learning curves

- Early stopping before validation error starts to increase

[Goodfellow, Bengio, Courville 2016]
Bagging

- Average multiple models trained on subsets of the data
- First subset: learns top loop, Second subset: bottom loop

[Goodfellow, Bengio, Courville 2016]
Dropout

- Random sample of connection weights is set to zero
- Train different network model each time
- Learn more robust, generalizable features

[Goodfellow, Bengio, Courville 2016]
Multitask learning

- Shared parameters are trained with more data
- Improved generalization error due to increased statistical strength

[Goodfellow, Bengio, Courville 2016]
Components of popular architectures
Convolution as edge detector

[Goodfellow, Bengio, Courville 2016]

Steven Bergner - CMPT 733
Gabor wavelets (kernels)

- Directional second derivative
- Second derivative (curvature)
- Local average, first derivative

[Goodfellow, Bengio, Courville 2016]
Gabor-like learned kernels

- Features extractors provided by pretrained networks

[Goodfellow, Bengio, Courville 2016]
Max pooling translation invariance

- Take max of certain neighbourhood
- Often combined, followed by downsampling

[Goodfellow, Bengio, Courville 2016]
Max pooling transform invariance

[Goodfellow, Bengio, Courville 2016]
Types of connectivity

Local connection: like convolution, but no sharing
Choosing architecture family

- No structure → fully connected
- Spatial structure → convolutional
- Sequential structure → recurrent
Software for Deep Learning
Current Frameworks

- Tensorflow / Keras
- PyTorch
- DL4J
- Caffe (superseded by Caffe2, which is merged into PyTorch)
  - And many more
- Most have CPU-only mode but much faster on NVIDIA GPU
Development strategy

- Identify needs: High accuracy or low accuracy?
- Choose metric
  - Accuracy (% of examples correct), Coverage (% examples processed)
  - Precision TP/(TP+FP), Recall TP/(TP+FN)
  - Amount of error in case of regression
- Build end-to-end system
  - Start from baseline, e.g. initialize with pre-trained network
- Refine driven by data
Sources