Deep Learning - COSC2779

Revision

Dr. Ruwan Tennakoon

RMIT UNIVERSITY

October 11, 2021

Course Structure

Week 1	Introduction Deep Learning	CNN
Week 2	Deep Feed Forward Networks	CNN
Week 3	Neural Network Optimization	CNN
Week 4	Convolutional Neural Networks	CNN
Week 5	Vision Application & CNN Architectures	CNN
Week 6	Practical methodology/Hyper parameter tuning	CNN
Week 7	Modelling Sequential (Time Series) Data	Time Series
Week 8	Time Series Applications	Time Series
Week 9	Unsupervised Learning/Generative Models	Advanced Topics
Week 10	Representation Learning/Self Supervised Learning	Advanced Topics
Week 11	Neural Network Model Interpretation/Explainable AI (XAI)	Advanced Topics
Week 12	Review	

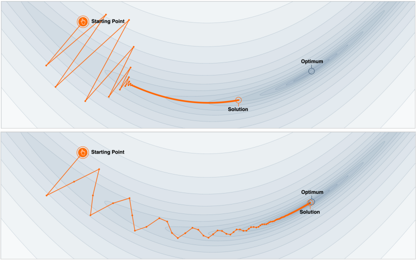
- Perceptron
- MLP
- Hidden units
- Output units
- Loss functions
- Universal approximation properties

- Single layer of a feed forward neural network learn a feature transformation: $\phi(\mathbf{x}) := h^{(i)}\left(\mathbf{x}; \mathbf{w}^{(i)}\right)$
- MLP: Model compose of many non linier feature transformations organized in a sequence (hierarchical). $h(\mathbf{x}) = h^{(3)}\left(h^{(2)}\left(h^{(1)}(\mathbf{x})\right)\right)$
- It is important for $h^{(i)}(\cdot)$ to have some non linearity. Stacking linear layers will still be linear.
- Activation functions add this non-linearity: Sigmoid, tanh, Relu, ...
- UAT: In theory one hidden layer MLP can approximate any function. In practice increasing depth helps.

- Back-prop
- Challenges in NN optimization
- Basic SGD
- Adaptive learning rate methods
- Parameter initialization
- Batch-Normalization

- In ML, We do not know p_{data}. Therefore we minimize the empirical risk. $\mathcal{L}(\mathbf{w}) = \underbrace{\mathbb{E}_{(\mathbf{x},y)\sim\hat{p}_{data}} \mathrm{L}(y, h(\mathbf{x};\mathbf{w}))}_{\text{Empirical Risk}} + \lambda \mathrm{R}(\mathbf{w})$
- Usually done with gradient based optimization techniques. Loss need to be differentiable but can be non-convex.
- Computing loss exactly is very expensive. In practice, we can compute these expectations on a small number of examples (batch) - SGD.
- Challanges: Local minimums, saddle points, cliffs, flat regions.
- Vanishing or exploding gradients: specially when network is very deep.

- Back-prop
- Challenges in NN optimization
- Basic SGD
- Adaptive learning rate methods
- Parameter initialization
- Batch-Normalization



Pathological curvature: regions of cost function which are not scaled properly.

- Momentum
- Adaptive Learning rate methods

- 2D convolution
- Pooling
- Variants of convolution

- In ML models for many tasks including computer vision we would like to have two properties:
  - Feature extraction usually happens locally - **sparse connectivity**.
  - In feature extraction the same operation is applied at different locations - **parameter sharing.**
- Both ideas can be achieved with convolutions
- Pooling help reduce redundant information and provide some level of invariance to translations.
- Variants of convolution: Strided convolution, dilated convolutions, transpose convolutions.

- Image Classification
- Object detection
- Segmentation

- AlexNet: First, most popular CNN for image classification. Several stages of Conv+pooling layers followed by MLP classifier.
- VGG: Smaller filters and deeper network.
- GoogLeNet: Multi resolution local topology, Global Average pooling (reduce parameters), Auxiliary classifiers for vanishing gradients.
- ResNet: Go very deep. Use skip connections to address vanishing gradients.
- Image segmentation/object detection can be seen as special cases of image classification.

- Practical methodology

- **Determine your goals**: Error metric and target value. Problem dependent.
- **Default Baseline Model**: Identify the components of end-to-end pipeline including - Baseline Models, cost functions, optimization.
- **Setup the diagnostic instrumentation**: Setup the visualizers and debuggers needed to determine bottlenecks in performance (overfitting/underfitting, weight changes, learned features, etc.).
- **Make incremental changes**: Repeatedly make incremental changes such as gathering new data, adjusting hyperparameters, or changing algorithms, based on specific findings from your instrumentation.

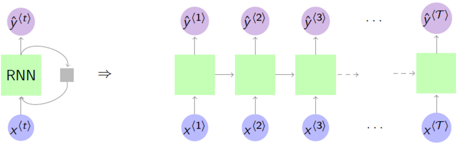
- Sequential data
- Recurrent neural networks
- Variants of RNN
- Bidirectional
- Deep

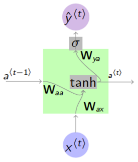
- Sequential data:
  - Data points with variable length .
  - Order of the data matters. Order can be:Time related: Video analysis, Not related to time: DNA sequence.
  - Shared feature across time is useful - context.
- Recurrent neural networks have loops in them, allowing information to persist.
- In theory, RNNs are absolutely capable of handling such "long-term dependencies". In practice,RNNs don't seem to be able to learn them. Solution: LSTM/GRU
- Modifications to RNN:
  - Bi-Directional: Use historical and future information.
  - Deep: More complex models.

Week 7: Sequential Data

Sequential Data:

- Data points with variable length.
- Order of the data matters
- Shared feature across time is useful.

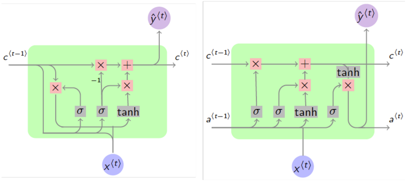




Simple RNN Cell

Improvements:

- Capture long range dependencies:



GRU LSTM

- Bi-Directional: Use both historical and future information.
- Deep RNN: Increase model capacity.

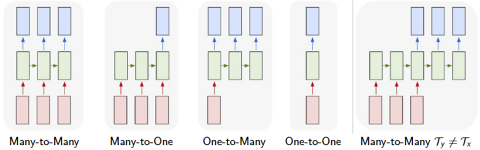


Image: The Unreasonable Effectiveness of Recurrent Neural Networks - Andrej Karpathy

- Text classification
- Machine translation
- Speech recognition

- Word embeddings convert the one-hot representation of a word to semantically meaningful feature vector. Usually learned on large related text corpus.
- Word2Vec and GloVe are more recent practical word embeddinglearning techniques
- Encoder decorder architecture enables machine translation. Works when one-to-one relationships does not hold.
- Attention provides a more advanced encorder decorder architecture.

- Transfer Learning
- Self Supervised Learning

How to build deep model when we do not have large labeled datasets.

- Transfer learning: Need pre-trained model on "similar" dataset + small amount of labeled data.
- Self-supervision: Need large unlabeled dataset.

Transfer Learning Process:

1. Select a related proxy task that has large dataset (or pre-trained model).
2. Train a selected base model on the proxy task or download the pre-trained model.
3. Identify the feature layer you want to use for your task.
4. Remove the layers above the bottleneck layer and reconfigure to your task (attach a head).
5. Freeze all the layer in base model.
6. Train the network on the new task.
7. (optional) Unfreeze some top layers in the base model and fine-tune.

- Deep generative models
- Auto Encoders
- GANs

Unsupervised learning: Deep generative models

- **AutoEncoders**: Interpretable latent space. Allows inference of $q(z|x)$, can be useful feature representation for other tasks. Samples blurrier and lower quality compared to state-of-the-art.
- **GANs**: Take game-theoretic approach. learn to generate from training distribution through 2-player game. Beautiful, state-of-the-art samples. Trickier & more unstable to train. Can't solve inference queries such as $p(x)$, $p(z|x)$.

A form of unsupervised learning.

Data:

$$\mathcal{D} = \left\{ \mathbf{x}^{(i)} \right\}_{i=1}^{N}$$

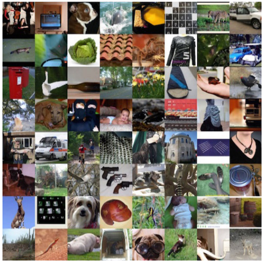
Goal: Given training data, generate new samples from same distribution.

Example: Autoencoders, GAN, One-to-Many RNN, ...

Probabilistic interpretation:

$$p\left(x_1^{(i)}, \cdots, x_d^{(i)}\right)$$

Learning:



$\longrightarrow$ $p_{model}(x)$

Inference (testing):

$p_{model}(x)$ $\longrightarrow$

Keeping up to Date

We covered many interesting topics this semester, and did not get a chance to cover many more. Hope you have now got the confidence to explore this evolving area of deep learning.

Useful resources:

- Deeplearning.ai has several interesting advanced deep courses (free): Link
- "The Batch" newsletter from Deeplearning.ai will inform you of some latest developments in the field.
- Be careful with blog posts. Papers from peer reviewed sources are more trustworthy (NIPS, ICML, CVPR, ICCV, ECCV, TPAMI, ...).

Be confident, you now know a lot about Deep Learning. Practice will make you a better DL engineer.

The End

Thank you for being a great class

Your feedback is very valuable to us. Please take few minutes (around 10 minutes) to complete the Course Experience Survey (CES).