UiO : Department of Informatics
University of Oslo

IN4310 Deep Learning for Image Analysis

Lecture 15: Course Recap

May 13, 2024

Dhananjay Tomar

Lecture 2: Linear Models

- Slides 14, 15, 32-35
- Empirical Risk Minimization (ERM)
- Gradient Descent for ERM
- Stochastic Gradient Descent for ERM
- Linear regression

$$\min_{\mathbf{w}} \left\{ R_n(\mathbf{w}) := \frac{1}{n} \sum_{i=1}^{n} \ell(h_{\mathbf{w}}(\mathbf{x}_i), \mathbf{y}_i) \right\}$$

$$h_{\mathbf{w}}(\mathbf{x}) = \mathbf{x}^\top \mathbf{w} = \sum_{i=1}^{d} x_i w_i, \mathbf{w} \in \mathbb{R}^d$$

$$h_{\mathbf{w},b}(\mathbf{x}) = \mathbf{x}^\top \mathbf{w} + b = \sum_{i=1}^{d} x_i w_i + b, \mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}$$

Lecture 3: Introduction to Neural Networks

- Slides 20-23, 43-44, 48, 63, 64, 67, 71, 72
- Logistic regression:
  - Assume we have a linear (or affine) mapping $f_{w,b}(x) = w \cdot x + b$. Plugging it into the logistic sigmoid function s(x) provides a logistic regression model:
    - $s(f_{w,b}(x)) = \frac{e^{w.x+b}}{1+e^{w.x+b}} = \frac{1}{1+e^{-w.x-b}}$
- Cross-entropy Loss Function

$$\mathbf{w}^\star = \arg\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} -y_i \log(s(h_{\mathbf{w}}(\mathbf{x}_i)) - (1-y_i)\log(1 - s(h_{\mathbf{w}}(\mathbf{x}_i)))$$

  - For one-hot labels $y_1, \ldots, y_n \in \{0, 1\}$, the cross-entropy loss of a data point (x_i, y_i) is the negative logarithm of the predicted probability of the ground-truth class.
    - If $y_i=1$, CE Loss = $-\log(s(f_{w,b}(x_i)))$
    - If $y_i=0$, CE Loss = $-\log(1 - s(f_{w,b}(x_i)))$

Lecture 3: Introduction to Neural Networks

- Artificial Neuron

$$z_j = \sum_{i=1}^{n} x_i w_{ij} + b_j$$

$$a_j = g(z_j) = g\left(\sum_{i=1}^{n} x_i w_{ij} + b_j\right)$$

- Hidden Layer

$$\mathbf{a}^{[l]} = g\left(\mathbf{W}^{[l]}\mathbf{a}^{[l-1]} + \mathbf{b}^{[l]}\right)$$

- Softmax function (output layer)

$$s(\mathbf{x})_k = \frac{e^{x_k}}{\sum_{i=1}^{n} e^{x_i}}$$

Lecture 4: Convolutional Neural networks

- Slides 13, 19, 23, 25, 33-35, 43
- 1-D and 2-D convolutions
- Output size of a convolution operation
- Stride
- Padding
- Receptive field
- Max Pooling, Average Pooling

Lecture 5: Deep Architecture Evolution

- Slides 19-21, 24-26, 38-40
- Dropout
- ResNets:
  - Residual / skip connections
    - Help gradients flow better, vanishing gradients
- Batch Normalization:
  - Step 1: Normalize activations of a layer by the running mean and running variance learnt at training time

$$\hat{\mathbf{x}} = \frac{\mathbf{x} - \mu_{\mathrm{run}}}{\sqrt{\sigma^2_{\mathrm{run}} + \epsilon}}$$

  - Step 2: apply **y = a$\hat{\mathbf{x}}$ + b** with **a, b** as the learnt rescaling parameters
- Group Normalization, Layer Normalization
- Finetuning

Lecture 6: Back Propagation and Optimization

- Slides 15-20, 28, 34, 35, 42, 43
- Back propagation:
  - Chain rule → calculating gradients in the backward pass
- Mini-batch
- SGD with Momentum
- Ada Grad, RMSProp, ADAM
  - An easy-to-read blog post to understand these optimizers: https://ruder.io/optimizing-gradient-descent/

Lecture 7: Performance estimation

- Slides 4-10, 15-16, 23-25, 28-30, 32, 44-70 (except 58, slides marked as non-relevant)
- Train, validation, test subsets
- External test set
- How to facilitate generalisation:
  - Control the network's capacity
  - Facilitate learning like with skip connections, transfer learning, optimizers
- Performance metric
  - Advantages & limitations of different metrics
- Uncertainty of performance estimate (only the basics)

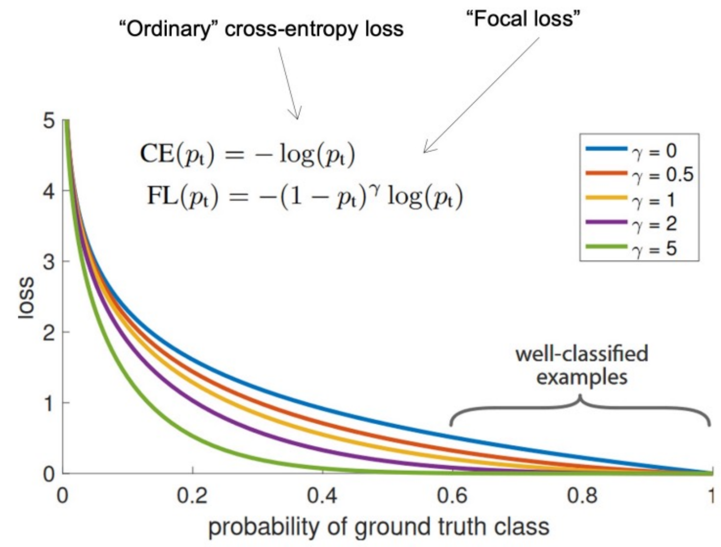
- No need to memorise the papers (with results) in the slides.
- No need to memorise the minor details in the slides.

Lecture 8: Data Augmentation

- Differences in Distributions
  - resolution, scale, lighting, colours, etc.
- Geometric transformations
  - Horizontal / vertical flips, rotation, random crop
- Photometric Transforms
  - ColorJitter, brightness, contrast, hue, saturation
- Other transformations
  - Blur, add noise, filters etc.
- Many other augmentations
- Weight / Parameter Initialization
  - Try to get 0 mean and same variance across all the layers
- Contrastive Learning

Lecture 9: Object Detection

- Slides 17-57
- One stage vs two stage detectors
- R-CNN, Fast R-CNN, Faster R-CNN (not very important but good to know)
- What are anchors and how are they used?
- How to handle different object sizes?
  – Use features from different layers that have different resolutions (different sized receptive fields)
  – SSD/FPN
- How to pick one box from multiple overlapping boxes?
  – Use Non-Maximum Suppression (NMS)
- How to match predictions with ground truth boxes during training?
- Focal Loss (Problem 3: Too many background predictions)
- Performance metrics used for object detection
  – Require setting an IoU treshold



Lecture 10: Image Segmentation

- Slides 1-28, 34-41
- Problem #1: How to capture global context?
  - Downsample feature maps
  - Use dilated / atrous convolution.
- Problem #2: How do we upsample features?
  - Nearest neighbour upsampling
  - Unpooling
  - Transposed convolution
- Problem #3: How to fetch precise boundary locations?
  - Feature Pyramid Network, UNet
  - Dilated / Atrous convolution
- Instance Segmentation
  - Mask R-CNN
- Performance metrics

- No need to memorise the exact architectures, just know the principles behind those architectures

Lecture 11: Adversarial examples

- White box attacks: The attacker has access to the model's parameters
- Targeted attacks: Trick the network to classify a sample into a fixed class which is different from the true class
- Untargeted attacks: Trick the network to misclassify the adversarial image
- Iterative gradient descent/ascent
  - Gradient ascent away from original class: untargeted
  - Gradient descent towards the least probable class: targeted
- Projected gradient method
- Defences
  - Adversarial training
  - Image transformations like quantisation
  - etc.
- Understand why do these attacks exist at all

Lecture 12: Recurrent Neural Networks

- Basics of RNNs
- Different input-output structures of RNNs: one-to-one, one-to-many, etc.
- Training RNNs: BPTT, truncated BPTT
- Exploding and vanishing gradients
  - Do not memorise the math in the slides
- Preserving long range dependencies: LSTM, GRU
  - I might ask you questions related to the architecture and how they help but I won’t ask you to draw the whole architecture or write all the equations.
- Multilayer RNNs, bidirectional RNNs

Lecture 13: Vision Transformers

- Attention mechanism
- Self-attention
  - Positional encodings
  - Masked self-attention
  - Multiple heads
- Transformer encoder and decoder blocks
  - You should know the difference but no need to memorise the exact architecture
- Using transformers for images
  - Vision transformer (ViT)
  - Swin transformer
- Object detection using transformers: DETR

Lecture 14: Distribution Shifts

- Slides 8-10, 12, 13, 20-22
- Types of Distribution Shifts
- Importance-weighted ERM

Let $f_{\mathbf{w}} : \mathcal{X} \to \mathcal{Y}$ denote a predictor parameterized by $\mathbf{w} \in \mathbb{R}^d$. IWERM:

$$\min_{\mathbf{w}} \left\{ \hat{\mathbb{E}}_{p^{\mathrm{tr}}(\mathbf{x},\mathbf{y})} \left[\frac{p^{\mathrm{te}}(\mathbf{x},\mathbf{y})}{p^{\mathrm{tr}}(\mathbf{x},\mathbf{y})} \ell(f_{\mathbf{w}}(\mathbf{x}), \mathbf{y}) \right\} \right]$$

- Expect questions to be some basic questions to see whether you understand the definitions of covariate and label shifts and the importance weighting definition and so on.