Deep Learning: Review Notes

January 3, 2024

Lecture 4: Linear Classification

- Soft-max Classifier (Multinomial Logistic Regression):

$$s_i = \frac{e^{o_i}}{\sum_{j=1}^{K} e^{o_i}}$$

- Loss function:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \ell(f(X_i, \theta), y_i)$$

- Cross-Entropy loss function: Let p_k be the predicted probability that the instance belongs to class k, and $p = (p_1, \ldots, p_K)$. Then

$$\ell(p, y) = -\sum_{k} I(y = k) \log p_k$$

- What it the connection of cross-entropy and maximum-likelihood estimator?
- What it the connection of cross-entropy and KL distance?

Lecture 5: Multi-Layer Perceptron

- Definition of Single layer network (perception model): $f_w(x) = \sigma(wx + b)$
- Show that perception model can only solve linear Separable problems. Provide some example of non Linear Separable Problems such as XOR.
- Definition of multi-layer network and related terminology: input layer, hidden layers, last layers, active function, neuron (node), Weights and Biases.
- Feedforward Algorithm:

$$f_W(x) = h_L \circ h_2 \circ \cdot \circ h_1(x)$$

where $h_i(x) = a(w_i x + b_i)/$

- Universal Approximation Theorem (for two layers and for Width-Bounded ReLU Networks).

Lecture 8: Back-propagation Algorithm

- Computational Graph and automatic differentiation.
- Back-propagation Algorithm:

$$\frac{\partial \ell(f_W(x), y)}{\partial h_i} = \frac{\partial \ell}{\partial h_L} \cdot \frac{\partial h_L}{\partial h_{L-1}} \cdot \dots \cdot \frac{\partial h_{i+1}}{\partial h_i}$$
$$\frac{\partial \ell}{\partial W} = \sum_i \frac{\partial \ell}{\partial h_i} \cdot \frac{\partial h_i}{\partial W}$$

Also, we know that

Downstream Gradient = Local Jackobian × Upstream Gradient

Lecture 9: Optimization

- Stochastic Gradient Descent (SGD), and convergence theorem.
- Classic Robbins Monro Condition: $\sum_{i=1}^{\infty} \eta_i = \infty, \sum_{i=1}^{\infty} \eta_i^2 < \infty$
- Comparing SGD, GD, and mini-batch
- Momentum and Nesterov Momentum. What is the intuition behind these two ideas?
- AdaGrad, RMSprop, Adam algorithms
- Second order optimization. Why is this impractical
- FBGS (optional)

Lecture 10:Convolutional Neural Networks

- Convolution operation on images defintion.
- Convolution layer, padding, stride, tensors. kernel, downsampling
- Output size formula:

  Input size: $C_{in} \times H \times W$, and C_{out}

  Hyperparameters:

  - Kernel size: $K_H \times K_W$
  - number of filters: C_{out}

- Padding: P, Stride: S
- filters of size $C_{in} \times K_H \times K_W$

Number of learnable parameters:
Weight matrix: $C_{out} \times C_{in} \times K_H \times K_W$
bias: C_{out}

Output size: $C_{out} \times H' \times W'$

$$H' = (H - K + 2P)/S + 1$$

$$W' = (W - K + 2P)/S + 1$$

Number of multiply operations: $C_{out} \times H' \times W' \times C_{in} \times K_H \times K_W$

- You do not need to memorize the architecture of different network. But it is good (optional) to know the names of some famous architecture: VGG19, Resnet, GoogleNet, DenseNet

Lecture 11: Training DNN

- Diffrent activation function: sigmoid, tanh, ReLu, LeakyRelu, ELU, GLU
- Why learning does not happened for saturated neurons? Why sigmoid is not a good choice?
- Compare sigmoid, tanh, Relu as activation function.
- What is the advantage of Leaky ReLU to ReLU?
- Why does initialization important?
- Show that $var_{input} = var_{output}$ if $varw = \frac{1}{\sqrt{n_{in}}}$? (LeCun Formula)
- What is Xaviar initialization? $var(w) = \frac{2}{\sqrt{n_{in}+n_{out}}}$
- What is batch Normalization?
- optional: How does batch normalization help?
- What are different regularization? Why does regularization important? (overfitting)
- What is Dropout? How does Dropout helps? (co-adaption, and ensambeling)