Deep Learning CS583B: Midterm Exam

October 23, 2022

Instructor: Jia Xu
Due: 5:30 PM, 24 October, 2020

Student name: Dhruv. Vaghela -

Student ID: _____

Student email address: _____

- **Read these instructions carefully**
- Fill-in your personal info as indicated above.
- There are 10 pages (including this one) and 10 questions.
- Fill in your answers with a clear handwritting. The answer will not be accepted if the scanned form of the writting is not easy to read.
- You can also direct edit PDF and fill in your answer.
- Submit your answer sheets on Cancas by 5:30 PM on the 24th.

good luck!

Thanks!

1 Question (10 points)

Suppose that you are training a Speech Recognition System on 100 million sentences. Your goal is to improve the accuracy of the translation system. Your small team of experts and engineers has 1 day only to improve the system. which of the following task/tasks would you?

1. Preprocessing
2. Change of the training data
3. Implement a new training algorithm
4. Add new features
5. Change pruning threshold in search

Justify your answer:

2 Question (10 points)

You are given X-ray images of patients where malignant tumors have been labeled at the pixel level (for each X-ray, you have another image where each pixel is labeled 1 for tumor and -1 for non tumor). Describe how you could use deep learning to solve the problem of detecting malignant tumors. specify which type of model you would use, and how you would go about setting up the dataset to train it.

1. Preprocessing - Data augmentation, cleaning of data. Balance the data for each classes. → Train, val, Test dataset.
2. Applying K-fold cross validation. Introducing randomness in the data makes the model robust.
3. ~~Architecture~~ - CNN (Image classification).
4. Batch Normalization
5. Hyper parameter tuning.
6. Introducing techniques to reduce overfitting. Pooling, Dropout etc.
7. Activation functions - ReLU.
8. Optimizers - SGD (learning rate, momentum).
   - ADAM.

---

(1) Input layer
(2) Conv2D
(3) Max Pooling
(4) Conv2D
(5) Max Pooling
(6) Dropout
(7) Dense (FC)
(8) Softmax

} Potential CNN Architecture.

3 Question (10 points)

Specify how convolution of a given image is computed. Assume the input is an image I of size $H \times W$ with C channels, the kernel K has size $N \times M$, the stride is $T \times S$, the operation performed is in fact cross-correlation (as usual in convolutional neural networks) and that O output channels are computed. Spell out the computations for both SAME and VALID padding schemes.

SAME - this adds the padding of 1 to the image.
- output size remains same as the input.

VALID - Filter window stays at valid position inside input map, so output size shrinks by 1.
No padding occurs.

Eg. x is the image.

x.shape = [2, 3] channel. 1.

VALID - no padding applied hence the shape is (1, 1).

SAME - We apply padding to image making shape [2, 4] (-inf and then apply max pooling) so output shape is [1, 2].

4 Question (10 points)

Consider a convolution layer. The input consists of 6 feature maps of size 20×20. The output consists of 8 feature maps, and the filters are of size 5×5. The convolution is done with a stride of 2 and zero padding, so the output feature maps are of size 10×10. For both parts, you can leave your expression as a product of integers; you do not need to actually compute the product. You do not need to show your work, but doing so can help you receive partial credit.

1. Determine the number of weights in this convolution layer.
2. Now suppose we made this a fully connected layer, but where the number of input and output units are kept the same as in the network described above. Determine the number of weights in this layer.

1 - There's one filter for each pair of an input and output feature map, and the filters are each 5×5. Therefore, the number of weights is 6×8×5×5 = 1200.

2 - There are 20 × 20 × 6 units in the input layer and 10 × 10 × 8 units in the output layer, so the number of weights is 20×20×6×10×10×8 = 1,920,000

- For a fully-connected deep network with one hidden layer, increasing the number of hidden units should have what effect on bias and variance? Explain briefly.

Increasing the no. of hidden units would make the model more complex. This would lead to lower bias and higher variance.

No. of hidden nodes $\propto \frac{1}{\text{Variance}}$.

- You are solving the binary classification task of classifying images as cat vs. non-cat. You design a CNN with a single output neuron. Let the output of this neuron be z. The final output of your network, $\hat{y}$ is given by: $\hat{y} = \delta(ReLU(z))$. You classify all inputs with a final value $\hat{y} \geq 0.5$ as cat images. What problem are you going to encounter?

Using ReLU then sigmoid will cause all predictions to be positive.

$(\sigma(\text{Relu}(z)) \geq 0.5 \quad \forall z)$.

5 Question

- A convolutional neural network has an input image of 28×28, and the filter is 3×3. The stride is 2. What is the output dimension of the 2nd layer?

Output d = $\left(\left\lfloor \frac{d_1 - K_1}{S} \right\rfloor + 1\right) \times \left(\left\lfloor \frac{d_2 - K_2}{S} \right\rfloor + 1\right) = \left(\frac{28-3}{2} + 1\right)^2 = \left(\frac{25}{2} + 1\right)^2 = (13 \times 13)$

- What is the max pooling result of a 2×2 pool size on the below input?

20	55	101	102
32	11	103	104
55	55	10	20
55	55	30	40

⇒

55	104
55	40

- What is the average pooling result of a 2×2 pool size on the above input?

29	102
55	25

6 Question

- Name a possible solution for the vanishing of the gradients and explain.

Use ReLu - like activation functions: ReLu activation functions keep linearity for regions where sigmoid and TanH are saturated, thus responding better to gradient vanishing / exploding. You can also use different types like Leaky-ReLu, Randomized ReLu, etc.

Use Batch Normalization (BN): this is another solution you could use in order to make your network more robust against gradient vanishing / exploding, especially if you are using sigmoid or TanH as activation functions. Actually, BN gives you more flexibility during the selection of the activation function for your network. The obtained architecture gets more robust at training, given that it is less prone to diverging due to initialization values or from higher learning rates.

Reduce learning rate: if you increase your learning rate without considering using a ReLu-like activation function and/or not using BN, your network can diverge during training much more easily. By reducing your learning rate you can reduce the chance of suffering vanishing / exploding gradients problem, but your network will take longer to learn. That is why the first two options are located first in the list.

Change your architecture: If you are using Convolutional Neural Networks, for example, and you are suffering from vanishing / exploding gradients, it might make sense to move to a new architecture like ResNETs. In comparison to other networks, these structures connect different layers between each other, i.e. the so-called skip connections, acting as gradient highways, allowing the gradient to flow between the different layers unhindered.

Use proper weight initialization: you could use, for example, Xavier initialization Xavier et al. to reduce the chance of suffering vanishing / exploding gradients. By itself this option does not guarantee you will resolve these issues, but it makes your network more robust when combined with other methods.

Gradient clipping: this can be used when having exploding gradient problem. Firsthand, we select a threshold value, and in case the value returned by the function of a gradient is greater than this threshold, we set it to a different value. You can check more info here.

7 Question

Consider the following linear auto-encoder with 1 input and 1 output: $\tilde{x} = w_2 w_1 x$, trained with the squared reconstruction error:

$L(W) = \frac{1}{P} \sum_{i=1}^{P} \frac{1}{2}(x^i - w_2 w_1 x^i)^2$

The scalar training samples have variance 1.

(a) What is the set of solutions (with 0 loss)?

(b) Does the loss have a saddle point? Where?

8 Question

A neural network has been encrypted on a device. You can access neither its architecture nor the values of its parameters. Is it possible to create an adversarial example to attack this network? Explain why.

Yes.

A number of black box attacks involve model extraction (see the next section) to create a local model, sometimes known as a substitute or surrogate model. Existing attacks are then executed against the local model to generate adversarial samples with the hope that these samples also evade the target model. This often works because of the phenomenon of attack transferability.

9 Question

You want to perform a regression task with the following dataset: $x^{(i)} \in R$ and $y^{(i)} \in R$, $i = 1, \cdots, m$ are the the ith example and output in the dataset, respectively. Denote the prediction for example i by $f(x^{(i)})$. Remember that for a given loss L we minimize the following cost function

$J = \frac{1}{m} \sum_{i=1}^{m} L(f(x^{(i)}), y^{(i)})$.

In this part we are deciding between using loss 1 and loss 2, given by:

$L_1(f(x^{(i)}), y^{(i)}) = |y^{(i)} - f(x^{(i)})|$,

$L_2(f(x^{(i)}), y^{(i)}) = (y^{(i)} - f(x^{(i)})^2$.

(a) Draw $L_1(x, 0)$ and $L_2(x, 0)$ versus $x \in R$ on the same plot.

(b) An outlier is a datapoint which is very different from other datapoints of the same class. Based on your plots, which method do you think works better when there is a large number of outliers in your dataset? Hint: Contributions of outliers to gradient calculations should be as small as possible.

L_1. The reason is it penalizes less for outliers. We would like to ignore outliers if possible.
When it isn't possible, using a loss func which penalizes outliers is more robust.

(c) "Using L_2 loss forces the weights of the network to end up small." Do you agree with this statement? Why/Why not?

False. In this case the residual will be forced to be sparse not the weights.

10 Question

You want to perform a classification task. You are hesitant between two choices: Approach A and Approach B. The only difference between these two approaches is the loss function that is minimized.

Assume that $x^{(i)} \in R$ and $y^{(i)} \in \{1, -1\}$, $i = 1, \ldots, m$ are the ith example and output label in the dataset, respectively. $f(x^{(i)})$ denotes the output of the classifier for the ith example. Recall that for a given loss L you minimize the following cost function:

$J = \frac{1}{m} \sum_{i=1}^{m} L(f(x^{(i)}), y^{(i)})$.

As we mentioned, the only difference between approach A and approach B is the choice

$$L_A(f(x^{(i)}), y^{(i)}) = max\{0, 1 - y^{(i)} f(x^{(i)})\}, \tag{1}$$

$$L_B(f(x^{(i)}), y^{(i)}) = \log_2^{(1+exp(-y^{(i)} f(x^{(i)})))}. \tag{2}$$

(i) Rewrite L_B in terms of the sigmoid function.

$L_B(f(x^{(i)}), y^{(i)}) = -\log_2(\sigma(y^i, x^i))$

(ii) You are given an example with $y^{(i)} = -1$. What value of $f(x^{(i)})$ will minimize L_B?

$f(x^{(i)}) = -\infty$

(iii) You are given an example with $y^{(i)} = -1$. What is the greatest value of $f(x^{(i)})$ that will minimize L_A?

$f(x^i) = -1$