CS410: Artificial Intelligence 2020 Fall
Homework 3: Neural Networks and Convolutional Neural Networks
Due date: 23:59:59 (GMT +08:00), November 24, 2020

Chunyu Xue 518021910698

1 Single-choice Questions

1. What does a neuron compute?
A. A neuron computes an activation function followed by a linear function ($z = Wx + b$)
B. A neuron computes a linear function ($z = Wx + b$) followed by an activation function
C. A neuron computes a function g that scales the input x linearly ($Wx + b$)
D. A neuron computes the mean of all features before applying the output to an activation function

Answer: B.

2. The tanh activation usually works better than sigmoid activation function for hidden units because the mean of its output is closer to zero, and so it centers the data better for the next layer. True/False?
A. True
B. False

Answer: A.

3. You are building a binary classifier for recognizing cucumbers ($y = 1$) vs. watermelons ($y = 0$). Which one of these activation functions would you recommend using for the output layer?
A. ReLU
B. Leaky ReLU
C. sigmoid
D. tanh

Answer: C.

4. You have built a network using the tanh activation for all the hidden units. You initialize the weights to relative large values, using np.random.randn(..,..)*1000. What will happen?
A. It doesn't matter. So long as you initialize the weights randomly gradient

descent is not affected by whether the weights are large or small.
B. This will cause the inputs of the tanh to also be very large, thus causing gradients to also become large. You therefore have to set α to be very small to prevent divergence; this will slow down learning.
C. This will cause the inputs of the tanh to also be very large, causing the units to be "highly activated" and thus speed up learning compared to if the weights had to start from small values.
D. This will cause the inputs of the tanh to also be very large, thus causing gradients to be close to zero. The optimization algorithm will thus become slow.

Answer: D.

5. Which of the following statements is true?
A. The deeper layers of a neural network are typically computing more complex features of the input than the earlier layers.
B. The earlier layers of a neural network are typically computing more complex features of the input than the deeper layers.

Answer: A.

6. During forward propagation, in the forward function for a layer l you need to know what is the activation function in a layer (Sigmoid, tanh, ReLU, etc.). During backpropagation, the corresponding backward function also needs to know what is the activation function for layer l, since the gradient depends on it. True/False?
A. True
B. False

Answer: A.

7. You have an input volume that is $15 \times 15 \times 8$, and pad it using "pad=2." What is the dimension of the resulting volume (after padding)?
A. $17 \times 17 \times 10$
B. $19 \times 19 \times 8$
C. $19 \times 19 \times 12$
D. $17 \times 17 \times 8$

Answer: B.

8. You have an input volume that is $32 \times 32 \times 16$, and apply max pooling with a stride of 2 and a filter size of 2. What is the output volume?
A. $15 \times 15 \times 16$
B. $16 \times 16 \times 8$
C. $16 \times 16 \times 16$
D. $32 \times 32 \times 8$

Answer: C.

9. You have an input volume that is $63 \times 63 \times 16$, and convolve it with 32 filters that are each 7×7, and stride of 1. You want to use a "same" convolution. What is the padding?

A. 1
B. 2
C. 3
D. 7

Answer: C.

10. The "sparsity of connections" is a benefit of using convolutional layers. What does this mean?
A. Regularization causes gradient descent to set many of the parameters to zero.
B. Each filter is connected to every channel in the previous layer.
C. Each activation in the next layer depends on only a small number of activations from the previous layer.
D. Each layer in a convolutional network is connected only to two other layers.

Answer: C.

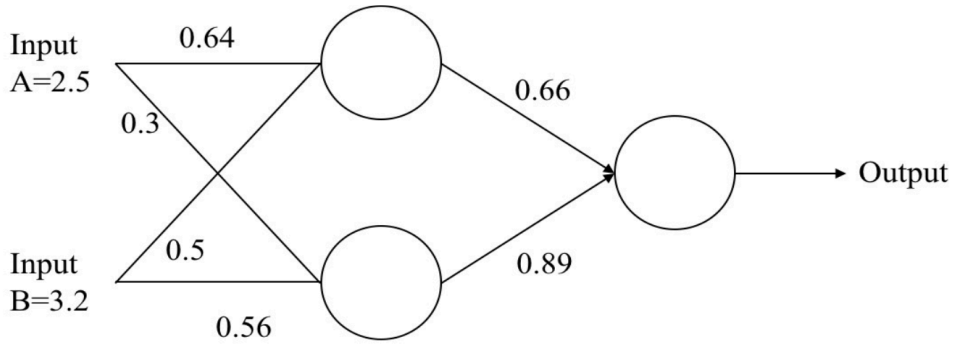
2 Calculation Questions (Please provide the detailed calculation process)

Question 1 (Neural Networks)
1. Retell the calculation process of backpropagation on Page 45 of the Lecture 6 slides.

Answer:

- Calculate errors of output neurons.
- Change output layer weights.
- Calculate (back-propagate) hidden layer errors.
- Change hidden layer weights.

2. Consider the simple network below:



(1). Assume that the neurons have sigmoid activation function
(2). Perform a forward pass on the network and find the predicted output
(3). Perform a reverse pass (training) once (target = 0.5) with $\eta = 1$
(4). Perform a further forward pass and comment on the result

Answer:

- Input part:
  Input to top neuron $= (2.5 \times 0.64) + (3.2 \times 0.5) = 3.2$. Out ≈ 0.96.
  Input to bottom neuron $= (3.2 \times 0.56) + (2.5 \times 0.3) = 2.542$. Out ≈ 0.927.
  Input to final neuron $= (0.66 \times 0.96) + (0.89 \times 0.927) \approx 1.459$. Out ≈ 0.811

- Output error:

  $\delta = (t - o) \times (1 - o) \times o = (0.5 - 0.811)(1 - 0.811)0.811 \approx -0.0477$

- New weights for output layer:

  $w_1^+ = w_1 + (\delta \times input) = 0.66 + (-0.0477 \times 0.96) \approx 0.614$

  $w_2^+ = w_2 + (\delta \times input) = 0.89 + (-0.0477 \times 0.927) \approx 0.846$

- Errors for hidden layers:

  $\delta_1 = \delta \times w_1 \times (1-o)o = -0.0477 \times 0.614 \times (1-0.96) \times 0.96 \approx -1.125 \times 10^{-3}$

  $\delta_2 = \delta \times w_2 \times (1-o)o = -0.0477 \times 0.846 \times (1-0.927) \times 0.927 \approx -2.731 \times 10^{-3}$

- New hidden layer weights:

  $w_3^+ = w_3 + (\delta_1 \times input) = 0.64 + (-1.125 \times 10^{-3} \times 2.5) \approx 0.637$

  $w_4^+ = w_4 + (\delta_1 \times input) = 0.5 + (-1.125 \times 10^{-3} \times 3.2) \approx 0.496$

  $w_5^+ = w_5 + (\delta_2 \times input) = 0.3 + (-2.731 \times 10^{-3} \times 2.5) \approx 0.293$

  $w_6^+ = w_6 + (\delta_2 \times input) = 0.56 + (-2.731 \times 10^{-3} \times 3.2) \approx 0.551$

- New input:

  Input to top neuron $= (2.5 \times 0.637) + (3.2 \times 0.496) \approx 3.180$. Out ≈ 0.96.

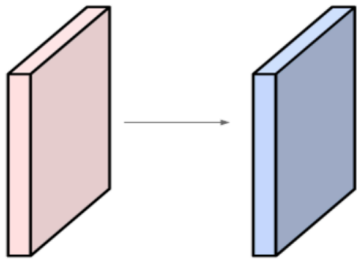
  Input to bottom neuron $= (3.2 \times 0.551) + (2.5 \times 0.293) = 2.496$. Out ≈ 0.924.

  Input to final neuron $= (0.614 \times 0.96) + (0.846 \times 0.924) \approx 1.371$. Out ≈ 0.798

Analysis:

The old error was -0.311, new error is -0.298. Therefore, the error has reduced.

Question 2 (Convolutional Neural Networks)



1. Suppose a convolutional layer with input volume $32 \times 32 \times 3$ and 10 5×5 filters with stride 1, pad 2.

(1). What is the output volume size of this layer? Show the computation steps of how to derive the volume size from the form layer.

Answer: The size calculation formula after convolution is:

$$N = \frac{W - F + 2P}{S} + 1 \tag{1}$$

By operating the the formula to the input width and height, we can get that the output width and height are both $\frac{32-5+2\times2}{1} + 1 = 32$.

Since there are 10 filters, the output volume size of this layer is $32 \times 32 \times 10$.

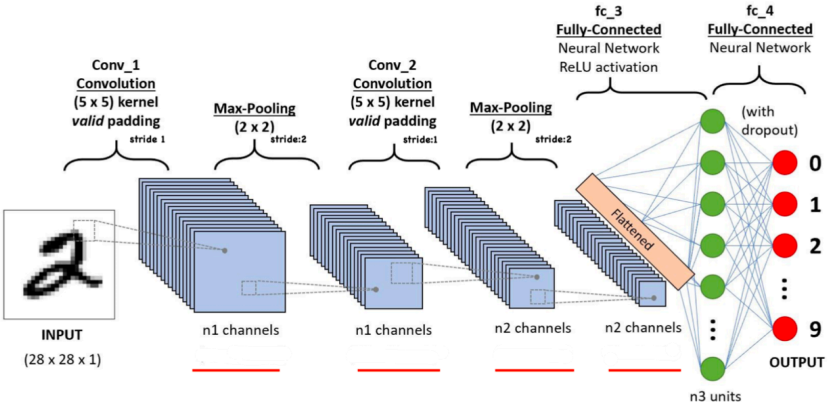
(2). What is the total number of parameters in this layer?

Answer: The calculation formula for number of parameters in the layer is:

Number of parameters = Size of kernel $\times$ Depth of the input volume $\times$ Number of kernels

Therefore, the total number of parameters in this layer is $5 \times 5 \times 3 \times 10 = 750$.

2. For the following LeNet-5 for the handwritten digit recognition task:



(1). Please calculate the output volume size of each layer (red line) and show the computation steps.

Answer: The size of input volume is $28 \times 28 \times 1$.

- $Conv_1$: This layer has n_1 5×5 convolution kernels, with $pad = 0$ and $stride = 1$.

$$N = \frac{W - F + 2P}{S} + 1 = \frac{28 - 5 + 2 \times 0}{1} + 1 = 24 \tag{1}$$

  Therefore, the output volume of this layer is $24 \times 24 \times n_1$.

- $MaxPooling_1$: The filter of this layer is 2×2, and $stride = 2$.

$$N = \frac{W - F}{S} + 1 = \frac{24 - 2}{2} + 1 = 12 \tag{2}$$

  Therefore, the output volume of this layer is $12 \times 12 \times n_1$.

- $Conv_2$: This layer has n_2 5×5 convolution kernels, with $pad = 0$ and $stride = 1$.

$$N = \frac{W - F + 2P}{S} + 1 = \frac{12 - 5 + 2 \times 0}{1} + 1 = 8 \tag{3}$$

  Therefore, the output volume of this layer is $8 \times 8 \times n_2$.

- $MaxPooling_2$: The filter of this layer is 2×2, and $stride = 2$.

$$N = \frac{W - F}{S} + 1 = \frac{8 - 2}{2} + 1 = 4 \tag{4}$$

Therefore, the output volume of this layer is $4 \times 4 \times n_2$.

(2). What is the total number of parameters in each layer?

Answer:

- $Conv_1$: This layer has $5 \times 5 \times 1 \times n_1 = 25n_1$ parameters.
- $MaxPooling_1$: This layer has no parameter.
- $Conv_2$: This layer has $5 \times 5 \times n_1 \times n_2 = 25n_1n_2$ parameters.
- $MaxPooling2$: This layer has no parameter.

3 Short Answer Questions

1. Try to summarize common activation functions and their characteristics.

Answer: Common activation functions and characteristics are as follows:

- **Sigmoid** function: Range from 0 to 1. Motivated by biological neurons and can be interpreted as the probability of an artificial neuron "firing" given its inputs. Sigmoid is commonly used in binary classification. The disadvantage is that the amount of calculation is large.
- **Tanh** function: Range from -1 to 1. Strongly negative inputs to the tanh will map to negative outputs, and only zero-valued inputs are mapped to near-zero outputs. Less likely to get "stuck" during training. The mean value of tanh is 0.
- **ReLU** function: Commonly used in hidden layer output. When the input is less than 0, the output will be 0; as long as the input is no less than 0, the output will be equal to the input. ReLU can alleviate the gradient decay problem, but will slowly cause the "Neuron Death".
- **Softmax** function: Commonly used in multi-classification neuron network output. Softmax uses polynomial distribution for modeling.

2. Summarize methods to avoid overfitting when training the neural network.

Answer: Methods to avoid overfitting are as follows:

- **Early stopping**. We record the accuracy of each epoch during the training, if the current accuracy is no better than the best accuracy for several times, we stop the training.
- **Data augmentation**. We enlarge the data set in certain ways.
- **Regularization**. We add a regularization term to the loss function to constraint on the weight coefficients.
- **Dropout**. We modify the structure of our neuron network to avoid overfitting.

3. Describe the motivation of batch normalization and analyze how this method solve the initial problem.

Answer: The motivation and solution are as follows:

- **Motivation:** When we train our neuron network, the tiny change on parameters in shallow layers will be enlarged by passing linear transforms and activation functions in multiple layers. Therefore, the input distribution in relatively deep layers will change and our model will be difficult to converge.

- **Solution:** Batch mormalization has introduced two "Reconstruction Parameters" that can be learned, which can reconstruct the learned features of the layer from the normalized data. By operating:
  - Calculate the mean value of batch data
  - Calculate the variance of batch data
  - Normalization
  - Scale transformation and migration

  We can significantly solve the initial problem and improve our model.