CS7150.37338.202130 Day 19 Preparation Questions

Aishwarya Vantipuli, Harshita Ved, Aveek Choudhury

TOTAL POINTS

5 / 5

QUESTION 1

1 Question 2 1 / 1

✓ **- 0 pts** **Correct**

QUESTION 2

2 Question 3 1 / 1

✓ **- 0 pts** **Correct**

- 1 pts Click here to replace this description.

Don't copy text from the paper. Explain how equation (3) is derived. z

QUESTION 3

3 Question 4 1 / 1

✓ **- 0 pts** **Correct**

QUESTION 4

4 Question 5 1 / 1

✓ **- 0 pts** **Correct**

- 0.5 pts Click here to replace this description.

QUESTION 5

5 Question 6 1 / 1

✓ **- 0 pts** **Correct**

CS 7150: Deep Learning — Spring 2021 — Paul Hand

Day 19 — Preparation Questions For Class
Due: Wednesday 3/31/2021 at 2:30pm via Gradescope

Names: **Aishwarya Vantipuli, Aveek Choudhury, Harshita Ved**

You may consult any and all resources in answering the questions. Your goal is to have answers that are ready to be shared with the class (or on a hypothetical job interview) as written. Your answers should be as concise as possible. When asked to explain a figure, your response should have the following structure: provide context (state what experiment was being run / state what problem is being solved), state what has been plotted, remark on what we observe from the plots, and interpret the results.

Submit one document for your group and tag all group members. We recommend you use Overleaf for joint editing of this TeX document.

Directions: Read 'Overcoming catastrophic forgetting in neural networks' by Kirkpatrick et al.

- Read Sections 1, 2.0, 2.1, 3

Question 1. *Provide a summary of the contributions of this paper.*

Response: Summary of "Overcoming catastrophic forgetting in neural networks" paper:

- Catastrophic forgetting is the tendency of an artificial neural network to completely and abruptly forget previously learned information upon learning new information.
- This paper shows their scalable approach to overcome this limitation i.e. to make continual learning possible without having to learn the joint distributions of two different tasks and rather allows for sequential learning of two tasks.
- This paper also sheds light on how our artificial networks can take inspiration or in some cases a direct implementation of biological neurological processes.
- This paper presented an algorithm elastic weight consolidation, determines weights crucial for a task, holds them for the next task such that our NN models work better on both tasks. The algorithm is run in a supervised and reinforcement learning environment.

Question 2. *Derive equation (2). Your response should point out any assumptions the derivation is making.*

Response: To derive Equation (2) of the paper, let us start with the notations below:

- θ - Set of weights and biases of the linear projections that needs adjusting in the learning task to optimize performance.

- D - The given data needed to find parameters equal to their best probable values from a probabilistic perspective.
- $p(\mathrm{D}|\theta)$ - Computed probability of the data D using Bayes' rule.
- $p(\theta|\mathrm{D})$ - The computed conditional probability from the prior probability of the parameters $p(\theta)$. Eq. 1 written below.

$$p(\theta|D) = p(D|\theta) * p(\theta)/p(D) \tag{1}$$

Now:

If we apply a log-transform, the equation (1) can be rewritten as follows:

$$\log p(\theta|D) = \log p(D|\theta) + \log p(\theta) - \log p(D) \tag{2}$$

Suppose that the data D is composed of two independent parts, data D_A for task A and data D_B for task B.

Assumption made here: The independence of the two split of data D as D_A and D_B. Is it acceptable to assume independence?? The derivation assumes it is.

Using the definition of independence, we can rewrite our equation 2:

$$\log p(\theta|D) = \log [p(D_A|\theta).p(D_B|\theta)] + \log p(\theta) - \log [p(D_A).p(D_B)] \tag{3}$$

$$\log p(\theta|D) = \log p(D_B|\theta) + \log p(D_A|\theta) + \log p(\theta) - \log p(D_A) - \log p(D_B) \tag{4}$$

The center three terms of equation 4 are equivalent to the log of the conditional probability of the network's parameters given task A's data. Hence Eq. 4 becomes below:

$$\log p(\theta|D) = \log p(D_B|\theta) + \log p(\theta|D_A) - \log p(D_B) \tag{5}$$

In Eq. 5 the left side is telling us how to calculate $p(\theta|D)$ for the entire dataset, but all the information learned when solving task A is contained in the conditional probability $p(\theta|D_A)$. This conditional probability can tell us which parameters are important in solving task A. This is the key key to implementing EWC.

The Equation 5 derived above is the Eq. 2 of paper required in this question to derive.

Question 3. *Explain how formulation (3) is obtained from equation (2).*

Response:

$\log p(\theta|D) = \log p(D_B|\theta) + \log p(\theta|D_A) - \log p(D_B)$ (formulation 2)

All the information about task A must therefore have been absorbed into the posterior distribution $p(\theta|D_A)$. This posterior probability must contain information about which parameters

1 Question 2 1 / 1

✓ - 0 pts Correct

- D - The given data needed to find parameters equal to their best probable values from a probabilistic perspective.
- $p(\mathrm{D}|\theta)$ - Computed probability of the data D using Bayes' rule.
- $p(\theta|\mathrm{D})$ - The computed conditional probability from the prior probability of the parameters $p(\theta)$. Eq. 1 written below.

$$p(\theta|D) = p(D|\theta) * p(\theta)/p(D) \tag{1}$$

Now:

If we apply a log-transform, the equation (1) can be rewritten as follows:

$$\log p(\theta|D) = \log p(D|\theta) + \log p(\theta) - \log p(D) \tag{2}$$

Suppose that the data D is composed of two independent parts, data D_A for task A and data D_B for task B.

Assumption made here: The independence of the two split of data D as D_A and D_B. Is it acceptable to assume independence?? The derivation assumes it is.

Using the definition of independence, we can rewrite our equation 2:

$$\log p(\theta|D) = \log [p(D_A|\theta).p(D_B|\theta)] + \log p(\theta) - \log [p(D_A).p(D_B)] \tag{3}$$

$$\log p(\theta|D) = \log p(D_B|\theta) + \log p(D_A|\theta) + \log p(\theta) - \log p(D_A) - \log p(D_B) \tag{4}$$

The center three terms of equation 4 are equivalent to the log of the conditional probability of the network's parameters given task A's data. Hence Eq. 4 becomes below:

$$\log p(\theta|D) = \log p(D_B|\theta) + \log p(\theta|D_A) - \log p(D_B) \tag{5}$$

In Eq. 5 the left side is telling us how to calculate $p(\theta|D)$ for the entire dataset, but all the information learned when solving task A is contained in the conditional probability $p(\theta|D_A)$. This conditional probability can tell us which parameters are important in solving task A. This is the key key to implementing EWC.

The Equation 5 derived above is the Eq. 2 of paper required in this question to derive.

Question 3. *Explain how formulation (3) is obtained from equation (2).*

Response:

$\log p(\theta|D) = \log p(D_B|\theta) + \log p(\theta|D_A) - \log p(D_B)$ (formulation 2)

All the information about task A must therefore have been absorbed into the posterior distribution $p(\theta|D_A)$. This posterior probability must contain information about which parameters

were important to task A and is therefore the key to implementing EWC. We approximate the posterioras a Gaussian distribution with mean given by the parameters A and a diagonal precision given by the diagonal of the Fisher information matrix F . This approach is similar to expectation propagation where each subtask is seen as a factor of the posterior. When moving to a third task, task C, EWC will try to keep the network parameters close to the learned parameters of both task A and B. This can be enforced either with two separate penalties, or as one by noting that the sum of two quadratic penalties is itself a quadratic penalty.

Question 4. *Explain Figure 2ab. Make sure to include the context, a statement of what literally is plotted, what is to be observed, and what is concluded.*

Response:

Context:

A fully connected multilayer neural network was trained on several supervised learning tasks, constructed from the MNIST written digits classification problem, in sequence in order to compare the performance of EWC against SGD and L2 regularization in terms of catastrophic forgetting. Within each task, the network was trained in a traditional way, namely by shuffling the data and processing in small batches. After a fixed amount of training on each task, no further training on that task's dataset was allowed.

What is plotted:

Fig 2(A) plots the performance on test sets by 3 algorithms - EWC, SGD and L2-regularization as a function over training time on a set of 3 random permutation tasks - A, B, C when the network is trained on the data sequentially.

Fig. 2(B) plots the average performance across all tasks by the network when using EWC and SGD with dropout regularization as the number of sequential tasks increases.

What we observe:

In fig. 2(A), we observe that as we move from training the network on one task to the next (for example from A to B and C), the performance of SGD on the older task (A) declines. Only EWC is able to show high performance on older tasks as well as current tasks as the training progresses. The L2-regularized network is able to perform within considerable range on the older tasks, but shows poor performance in the newer tasks.

In fig. 2(B), we observe that EWC shows a rather stable performance as the number of sequential tasks increases with a slight dip towards the end whereas SGD+dropout shows a considerably steeper decrease in performance as the number of tasks increases.

Interpretation:

When training on a sequence of tasks with SGD+dropout incurs catastrophic forgetting, i.e. as the training shifts from one task to another, the algorithm fails to perform well on the older task. Even regularizing the network with a fixed quadratic constant can't counter this problem as the constraint protects all weights equally, leaving little spare capacity for learning the current tasks - as a result of which L2-regularization fails to perform well on the current task. EWC is able to show consistently good performance on all tasks even as we switch training tasks.

2 Question 3 1 / 1

✓ **- 0 pts** **Correct**

- 1 pts Click here to replace this description.

Don't copy text from the paper. Explain how equation (3) is derived. z

were important to task A and is therefore the key to implementing EWC. We approximate the posterioras a Gaussian distribution with mean given by the parameters A and a diagonal precision given by the diagonal of the Fisher information matrix F . This approach is similar to expectation propagation where each subtask is seen as a factor of the posterior. When moving to a third task, task C, EWC will try to keep the network parameters close to the learned parameters of both task A and B. This can be enforced either with two separate penalties, or as one by noting that the sum of two quadratic penalties is itself a quadratic penalty.

Question 4. *Explain Figure 2ab. Make sure to include the context, a statement of what literally is plotted, what is to be observed, and what is concluded.*

Response:

Context:

A fully connected multilayer neural network was trained on several supervised learning tasks, constructed from the MNIST written digits classification problem, in sequence in order to compare the performance of EWC against SGD and L2 regularization in terms of catastrophic forgetting. Within each task, the network was trained in a traditional way, namely by shuffling the data and processing in small batches. After a fixed amount of training on each task, no further training on that task's dataset was allowed.

What is plotted:

Fig 2(A) plots the performance on test sets by 3 algorithms - EWC, SGD and L2-regularization as a function over training time on a set of 3 random permutation tasks - A, B, C when the network is trained on the data sequentially.

Fig. 2(B) plots the average performance across all tasks by the network when using EWC and SGD with dropout regularization as the number of sequential tasks increases.

What we observe:

In fig. 2(A), we observe that as we move from training the network on one task to the next (for example from A to B and C), the performance of SGD on the older task (A) declines. Only EWC is able to show high performance on older tasks as well as current tasks as the training progresses. The L2-regularized network is able to perform within considerable range on the older tasks, but shows poor performance in the newer tasks.

In fig. 2(B), we observe that EWC shows a rather stable performance as the number of sequential tasks increases with a slight dip towards the end whereas SGD+dropout shows a considerably steeper decrease in performance as the number of tasks increases.

Interpretation:

When training on a sequence of tasks with SGD+dropout incurs catastrophic forgetting, i.e. as the training shifts from one task to another, the algorithm fails to perform well on the older task. Even regularizing the network with a fixed quadratic constant can't counter this problem as the constraint protects all weights equally, leaving little spare capacity for learning the current tasks - as a result of which L2-regularization fails to perform well on the current task. EWC is able to show consistently good performance on all tasks even as we switch training tasks.

Also, as the number of sequential tasks increases, EWC is able to show performance comparable to a single task network.

Question 5. *Explain Figure 2c. Make sure to include the context, a statement of what literally is plotted, what is to be observed, and what is concluded.*

Response:

Context:

To address the problem of whether elastic weight consolidation could allow deep neural networks to learn a set of complex tasks without catastrophic forgetting, authors trained a fully connected multi layer neural network on several supervised learning tasks in sequence by y by shuffling the data and processing it in small batches using MNIST.

What is plotted:

Figure shows Similarity between the Fisher information matrices as a function of network depth for two different amounts of permutation. Small square of 8x8 pixels in the middle of the image is permuted (grey) and large square of 26x26 pixels is permuted (black).

What we observe:

when a network is trained on two tasks which are very similar to each other the tasks depend on similar sets of weights throughout the whole network (grey curve). When then the two tasks are more dissimilar from each other, the network begins to allocate separate capacity for the two tasks (black line).

Interpretation:

More different the tasks are, the smaller the overlap in Fisher information matrices in early layers.Even for the large permutations, the layers of the network closer to the output are indeed being reused for both tasks. This reflects the fact that the permutations make the input domain very different, but the output domain (i.e. the class labels) is shared.

Question 6. *How is the algorithm in this paper biologically inspired? Why is the method called 'elastic weight consolidation'?*

Response:

The ability to learn tasks in succession without forgetting is a core component of biological and artificial intelligence. EWC allows knowledge of previous tasks to be protected during new learning, thereby avoiding catastrophic forgetting of old abilities. It does so by selectively decreasing the plasticity of weights, and thus has parallels with neurobiological models of synaptic consolidation.

While learning a task, EWC therefore protects the performance in previous task by constraining the parameters to stay in a region of low error for that task. This constraint is implemented as a quadratic penalty, and can therefore be imagined as a spring anchoring the parameters to the previous solution, hence the name elastic.

3 Question 4 1 / 1

✓ - 0 pts Correct

Also, as the number of sequential tasks increases, EWC is able to show performance comparable to a single task network.

Question 5. *Explain Figure 2c. Make sure to include the context, a statement of what literally is plotted, what is to be observed, and what is concluded.*

Response:

Context:

To address the problem of whether elastic weight consolidation could allow deep neural networks to learn a set of complex tasks without catastrophic forgetting, authors trained a fully connected multi layer neural network on several supervised learning tasks in sequence by y by shuffling the data and processing it in small batches using MNIST.

What is plotted:

Figure shows Similarity between the Fisher information matrices as a function of network depth for two different amounts of permutation. Small square of 8x8 pixels in the middle of the image is permuted (grey) and large square of 26x26 pixels is permuted (black).

What we observe:

when a network is trained on two tasks which are very similar to each other the tasks depend on similar sets of weights throughout the whole network (grey curve). When then the two tasks are more dissimilar from each other, the network begins to allocate separate capacity for the two tasks (black line).

Interpretation:

More different the tasks are, the smaller the overlap in Fisher information matrices in early layers.Even for the large permutations, the layers of the network closer to the output are indeed being reused for both tasks. This reflects the fact that the permutations make the input domain very different, but the output domain (i.e. the class labels) is shared.

Question 6. *How is the algorithm in this paper biologically inspired? Why is the method called 'elastic weight consolidation'?*

Response:

The ability to learn tasks in succession without forgetting is a core component of biological and artificial intelligence. EWC allows knowledge of previous tasks to be protected during new learning, thereby avoiding catastrophic forgetting of old abilities. It does so by selectively decreasing the plasticity of weights, and thus has parallels with neurobiological models of synaptic consolidation.

While learning a task, EWC therefore protects the performance in previous task by constraining the parameters to stay in a region of low error for that task. This constraint is implemented as a quadratic penalty, and can therefore be imagined as a spring anchoring the parameters to the previous solution, hence the name elastic.

4 Question 5 1 / 1

✓ **- 0 pts Correct**

- 0.5 pts Click here to replace this description.

Also, as the number of sequential tasks increases, EWC is able to show performance comparable to a single task network.

Question 5. *Explain Figure 2c. Make sure to include the context, a statement of what literally is plotted, what is to be observed, and what is concluded.*

Response:

Context:

To address the problem of whether elastic weight consolidation could allow deep neural networks to learn a set of complex tasks without catastrophic forgetting, authors trained a fully connected multi layer neural network on several supervised learning tasks in sequence by y by shuffling the data and processing it in small batches using MNIST.

What is plotted:

Figure shows Similarity between the Fisher information matrices as a function of network depth for two different amounts of permutation. Small square of 8x8 pixels in the middle of the image is permuted (grey) and large square of 26x26 pixels is permuted (black).

What we observe:

when a network is trained on two tasks which are very similar to each other the tasks depend on similar sets of weights throughout the whole network (grey curve). When then the two tasks are more dissimilar from each other, the network begins to allocate separate capacity for the two tasks (black line).

Interpretation:

More different the tasks are, the smaller the overlap in Fisher information matrices in early layers.Even for the large permutations, the layers of the network closer to the output are indeed being reused for both tasks. This reflects the fact that the permutations make the input domain very different, but the output domain (i.e. the class labels) is shared.

Question 6. *How is the algorithm in this paper biologically inspired? Why is the method called 'elastic weight consolidation'?*

Response:

The ability to learn tasks in succession without forgetting is a core component of biological and artificial intelligence. EWC allows knowledge of previous tasks to be protected during new learning, thereby avoiding catastrophic forgetting of old abilities. It does so by selectively decreasing the plasticity of weights, and thus has parallels with neurobiological models of synaptic consolidation.

While learning a task, EWC therefore protects the performance in previous task by constraining the parameters to stay in a region of low error for that task. This constraint is implemented as a quadratic penalty, and can therefore be imagined as a spring anchoring the parameters to the previous solution, hence the name elastic.

5 Question 6 1 / 1

✓ - 0 pts Correct