Overcoming Catastrophic Forgetting in Neural Networks

by James Kirkpatrick and DeepMind, 2016
Agenda

1. Motivation and introduction
2. Mathematics behind EWC
3. Experiments and results
   1. Random patterns
   2. MNIST
   3. Atari 2400
4. Alternative approaches
Motivation

• **Catastrophic forgetting** is a problem which occurs due to Neural Networks tendencies while learning new tasks to quickly overwrite and lose parameters necessary to perform well at a previous task.
Motivation

How is the mammalian brain capable of achieving this?
Motivation

Can catastrophic forgetting be overcome by using a similar approach of varying individual neuron's plasticity depending on each neuron's importance?

YES
Intuition

- Low error for task B
- Low error for task A
- EWC
- $L_2$
- no penalty

Maksym Sukhorukov
Mathematics

• What does training network mean from probabilistic perspective?

\[ p(\theta | D) = \frac{p(D | \theta)p(\theta)}{p(D)} \]

• Or applying log transform:

\[ \log p(\theta | D) = \log p(D | \theta) + \log p(\theta) - \log p(D) \]
Mathematics

• Suppose that the data $D$ are really composed of two independent parts, data $D_A$ and data $D_B$.

  • $\log p(\theta | D) = \log p(D_A | \theta)p(D_B | \theta) + \log p(\theta) − \log p(D_A)p(D_B) =$

  • $= \log p(D_B | \theta) + \log p(D_A | \theta) + \log p(\theta) − \log p(D_A) − \log p(D_B)$

• So we reach following:

  • $\log p(\theta | D) = \log p(D_B | \theta) + \log p(\theta | D_A) − \log p(D_B)$
Mathematics

• Bayesians never forget (catastrophically)

• We want to maintain full Bayesian posterior distribution:

\[ p(\theta \mid D_{T_1}, \ldots D_{T_k}) \]

• Then we could simply use this posterior as a prior when solving the task \( T_{k+1} \) and obtain an updated posterior which captures all tasks \( T_1 \ldots T_{k+1} \)
Mathematics

- \( \log p(\theta|D) = \log p(D_B|\theta) + \log p(\theta|D_A) - \log p(D_B) \)
- But \( \log p(\theta|D_A) \) is intractable. Why?
- \( p(D_A) = \int p(D_A|\theta')p(\theta')d\theta' \) where \( \theta' \) is a possible configuration of parameters in the parameter space.
Mathematics

• We need the mode $\arg\max_{\theta} \log p(\theta|D_A)$ and the Hessian of $\log p(\theta)$, evaluated at the mode.
• The mode $\theta^A$ can be found via optimizing the loss $L_A(\theta)$.
• The Hessian would be sum of the Fisher information from task A: $F^A$
Mathematics

Fisher Information Matrix:

aka 2nd derivative of likelihood
curvature function b/c it tells how curved
the likelihood func. is around maximum

\[ \mathcal{L}(\theta) \approx \mathcal{L}(\theta_{opt}) + \frac{\partial \mathcal{L}}{\partial \theta}{\bigg|}_{\theta_{opt}} (\theta - \theta_{opt}) \]

\[ + \frac{1}{2} \frac{\partial^2 \mathcal{L}}{\partial \theta^2}{\bigg|}_{\theta_{opt}} (\theta - \theta_{opt})^2 \]

assume gaussian so take log of 2nd derivative
and break up into weights

\[ \frac{\partial \ln \mathcal{L}}{\partial \theta_1} \frac{\partial \ln \mathcal{L}}{\partial \theta_2} \ldots = -F \]

for gaussian case, inverse of F is covariance matrix,
which tells us how well we have constrained
the parameters (weights) we are interested in.

Maksym Sukhorukov
Mathematics

- $\log p(\theta|D_A, D_B) \approx -L_B(\theta) - \frac{\lambda}{2} \sum_i F_{i,i}^A (\theta - \theta_i^A)^2 + \text{constant}$, where $L_B$ is a negative log likelihood, or loss function of task B and the Gaussian approximation to $p(\theta|D_A, D_B)$ acts as a nice quadratic regularizer, which is dependent on data from task A

- $\theta^{A,B} = \arg\min_{\theta} L_B(\theta) + \frac{\lambda}{2} \sum_i F_{i,i}^A (\theta_i - \theta_i^A)^2$
When moving to a third task, task C, EWC will try to keep the network parameters close to the learned parameters of both tasks 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.
Experiments and results. Random patterns
Experiments and results. Random patterns
Experiments and results. Random patterns
Experiments and results. MNIST
Experiments and results. MNIST
Experiments and results. Atari 2600

Maksym Sukhorukov
Experiments and results. Atari 2600

Maksym Sukhorukov
Experiments and results. Atari 2600

Maksym Sukhorukov
Experiments and results. Atari 2600
Experiments and results. Atari 2600

Maksym Sukhorukov
Experiments and results. Atari 2600

Maksym Sukhorukov
Experiments and results. Atari 2600
Experiments and results. Atari 2600

Maksym Sukhorukov
Network Overlap

low % permutation

high % permutation

Overlap in Fisher vs. Layer depth
Alternative approaches

• An **ensemble of DNNs**. For every new task, a new network is used with a shared representation from the previous task. This approach actually works quite well for all of the tasks but we can never expect this to scale for a large set of different tasks (for training and especially inference).
Alternative approaches

• A few of the recent approaches include PathNet which uses an evolutionary approach to deal with the forgetting. In the PathNet, each DNN can have ~20 modules for a given layer and a particular task may choose ~4 modules in each layer. This is sort of like an extension to the ensemble of models but it solves the increasing complexity issue as the number of tasks increase.
Alternative approaches

• The JMT model uses successive regularization terms for each tasks loss in order to combat the forgetting. The successive regularization uses the embedding parameters of the previous task at the current epoch and the previous epoch.

JMT Objectives:

$$J_1(\theta_{POS}) = - \sum_s \sum_t \log p(y_t^{(1)}) = \alpha|h_t^{(1)}) + \lambda W_{POS}|^2 + \delta||\theta - \theta'||^2$$
Final Interpretation

"Formally, when there is a new task to be learned, the network parameters are tempered by a prior which is the posterior distribution on the parameters given data from the previous task(s). This enables fast learning rates on parameters that are poorly constrained by the previous tasks and slow learning rates for those that are crucial."
Summary

• Learning tasks in succession without forgetting is necessary for intelligence
• Research shows synaptic consolidation in mammalian brains helps achieve continual learning
• EWC mimics synaptic consolidation in artificial neural networks by making important parameters less plastic
• EWC as applied in neural networks displays superior performance to SGD and uniform parameter stability in a range of domains
• EWC provides evidence that weight consolidation is fundamental to continual learning
Thank you for attention!