Overcoming Catastrophic Forgetting in Continual Learning of Neural Networks

Mehrdad Farajtabar
farajtabar@google.com

Google DeepMind

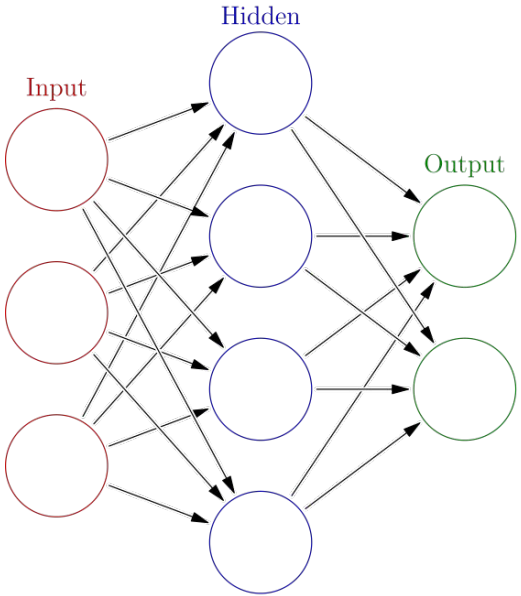
August 2020

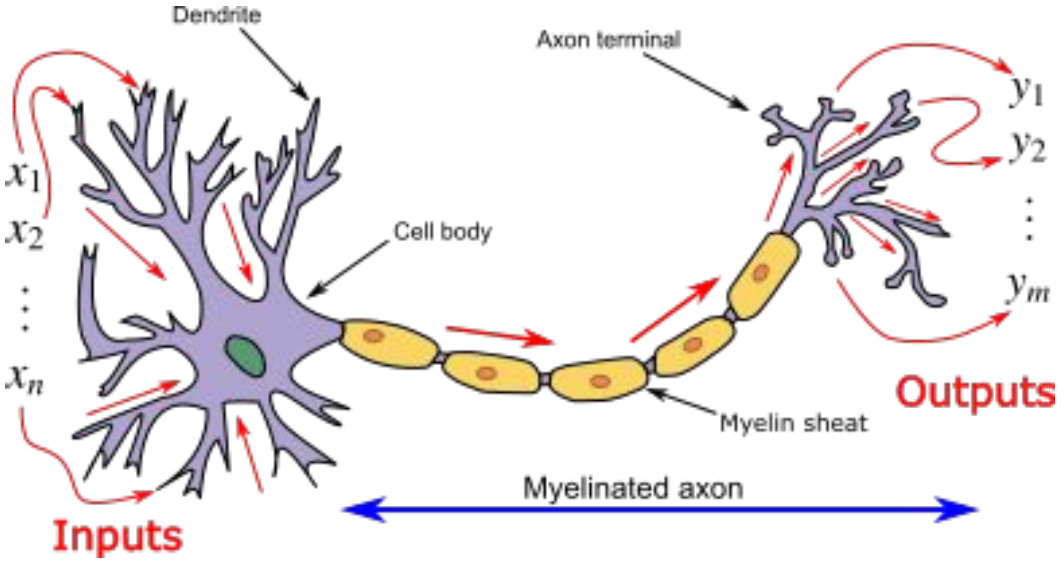
Outline

- Introduction to continual learning
- The problem of catastrophic forgetting
- Orthogonalization and regularization
- Dropout and implicit gating
- Training regimes and stabilization
- Taylor expansion and loss approximation

Artificial General Intelligence (AGI)

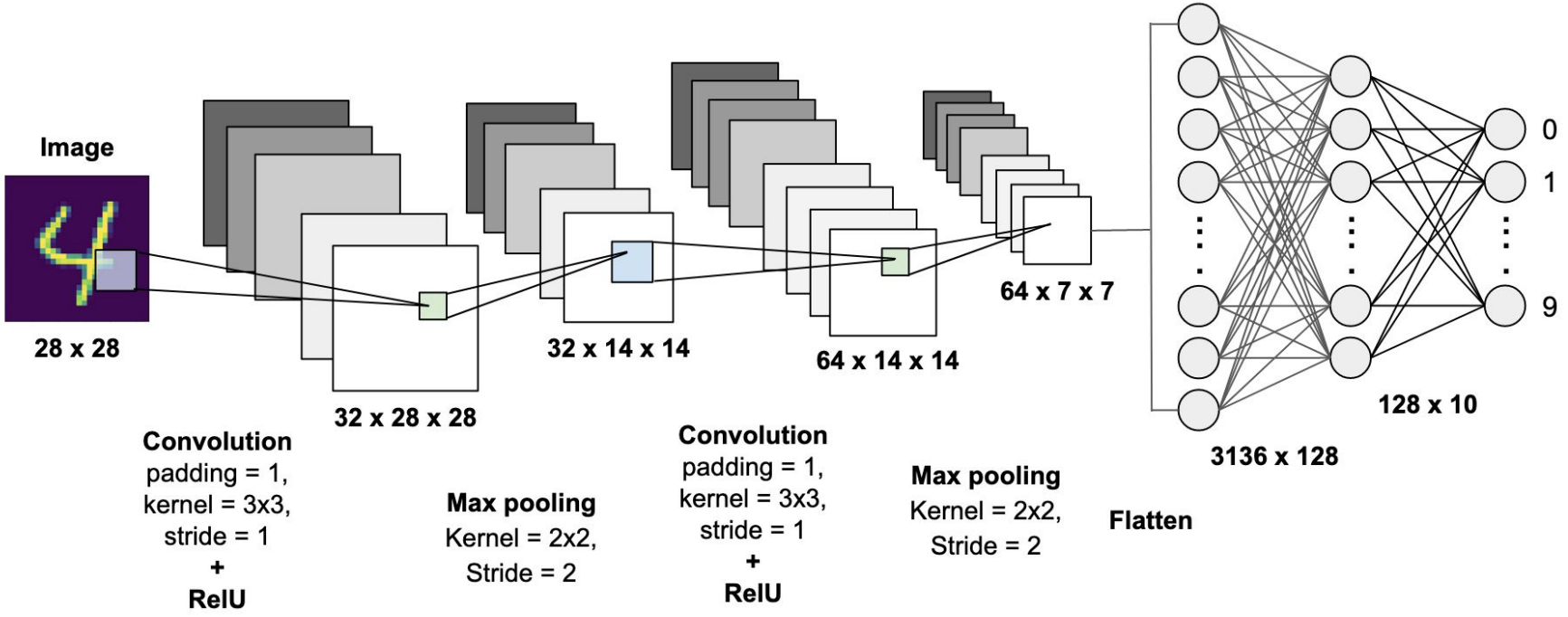
Artificial neural networks





Images from Wikipedia

Running example: handwritten digit recognition



Images from towardsdatascience.com

Real world machine learning

Real World

Time or Environments or Context

Machine Learning

Eirik Solheim / Via funnyjunk.com

North

South

Scene classification in a road trip

Continual learning

- Critical component of intelligence

Image from economist.com

Continual learning

- Critical component of intelligence
- New information is constantly available

Image from economist.com

Continual learning

- Critical component of intelligence
- New information is constantly available
- A sequence of different tasks

Image from economist.com

Catastrophic forgetting

- Continual Learning problem

Catastrophic forgetting

- **Continual Learning problem:**
  - Better adaptation and faster learning of new tasks

Catastrophic forgetting

- **Continual Learning problem:**
  - Better adaptation and faster learning of new tasks
  - Improving performance on the old tasks

Catastrophic forgetting

- **Continual Learning problem:**
  - Better adaptation and faster learning of new tasks
  - Improving performance on the old tasks
  - Task identification and boundary detection

Catastrophic forgetting

- **Continual Learning problem:**
  - Better adaptation and faster learning of new tasks
  - Improving performance on the old tasks
  - Task identification and boundary detection
  - Catastrophic forgetting

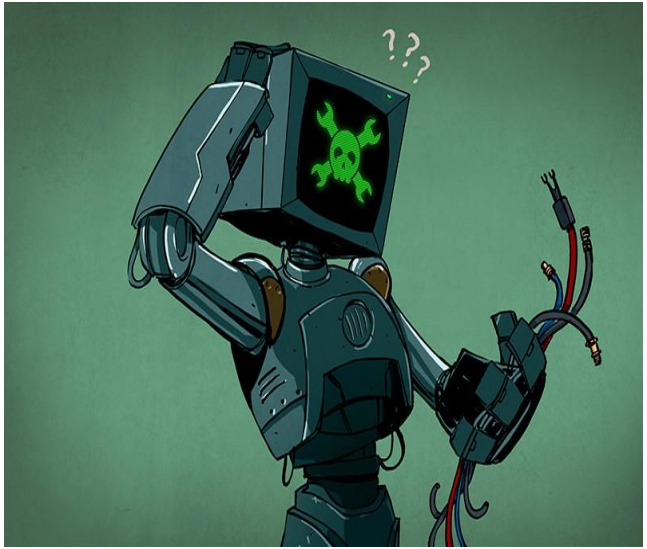


Image from hackday.com

Catastrophic forgetting

- **Continual Learning problem:**
  - Better adaptation and faster learning of new tasks
  - Improving performance on the old tasks
  - Task identification and boundary detection
  - **Catastrophic forgetting**

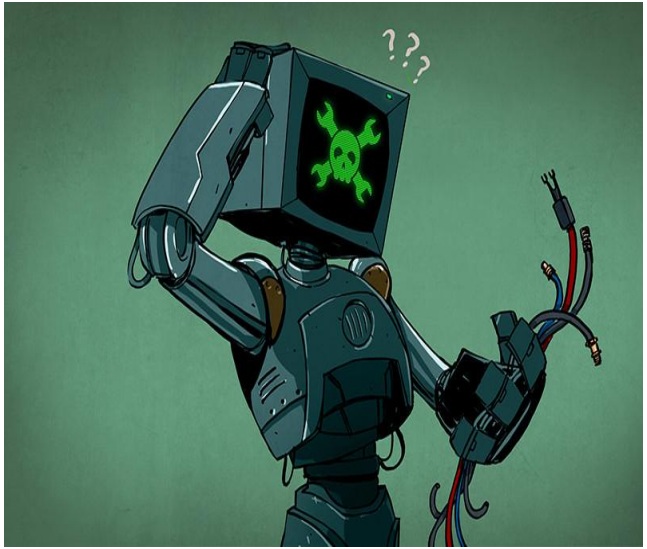
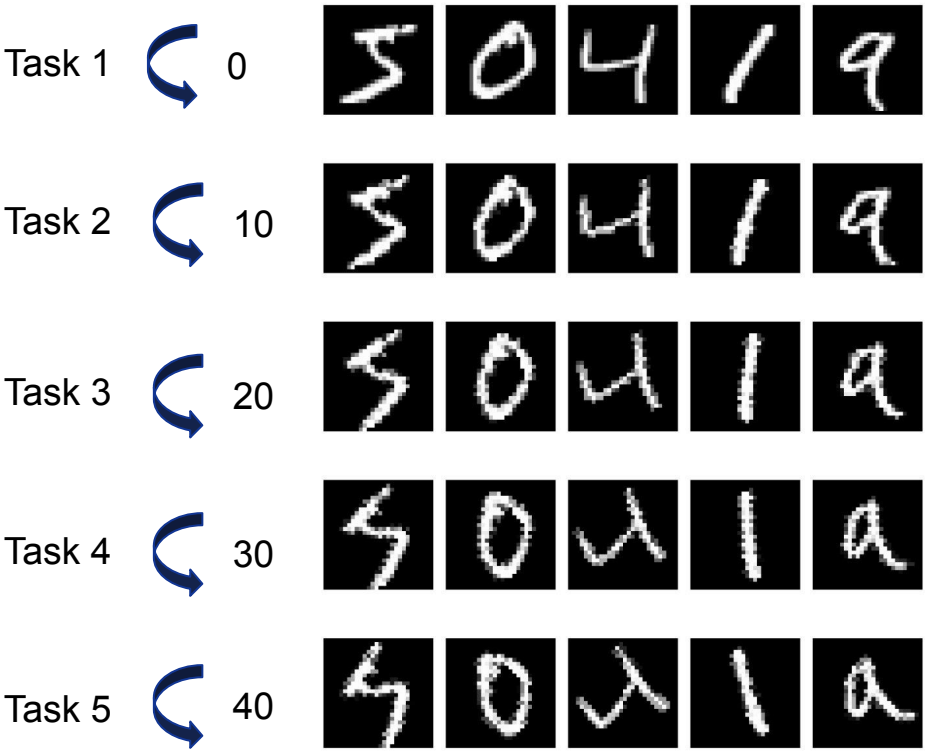


Image from hackday.com

Is forgetting really catastrophic?

Handwritten Digit Recognition

Rotated MNIST Benchmark



Side question: does it easily generalize?

A two layer MLP with 100 neurons trained on task 1

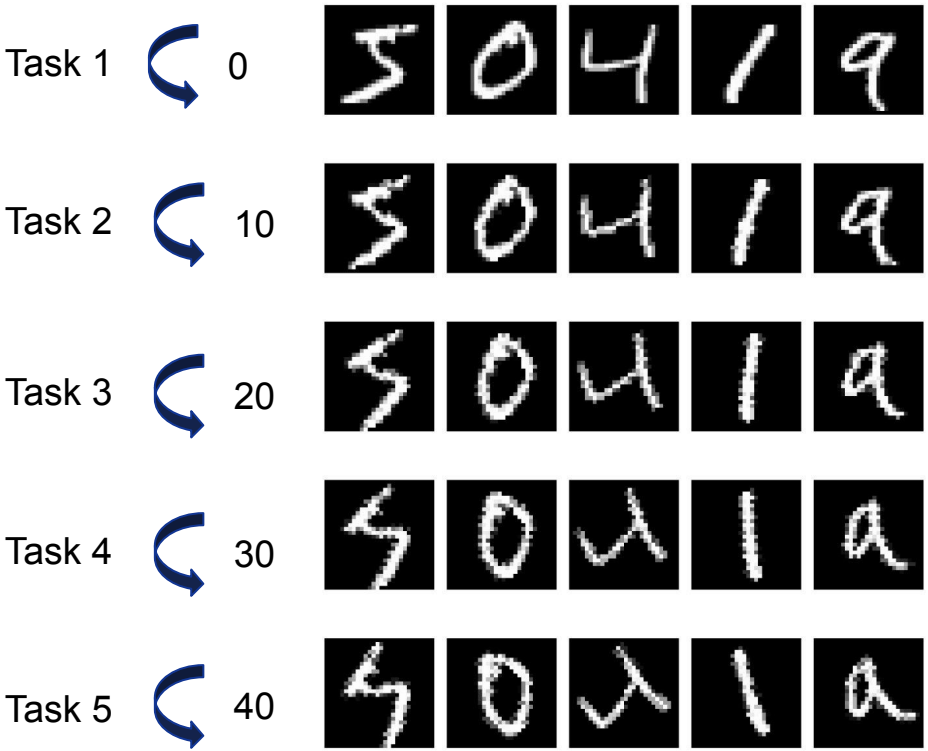
Test accuracy on task 1 > 94%

Test accuracy on task 4 ~ 50%

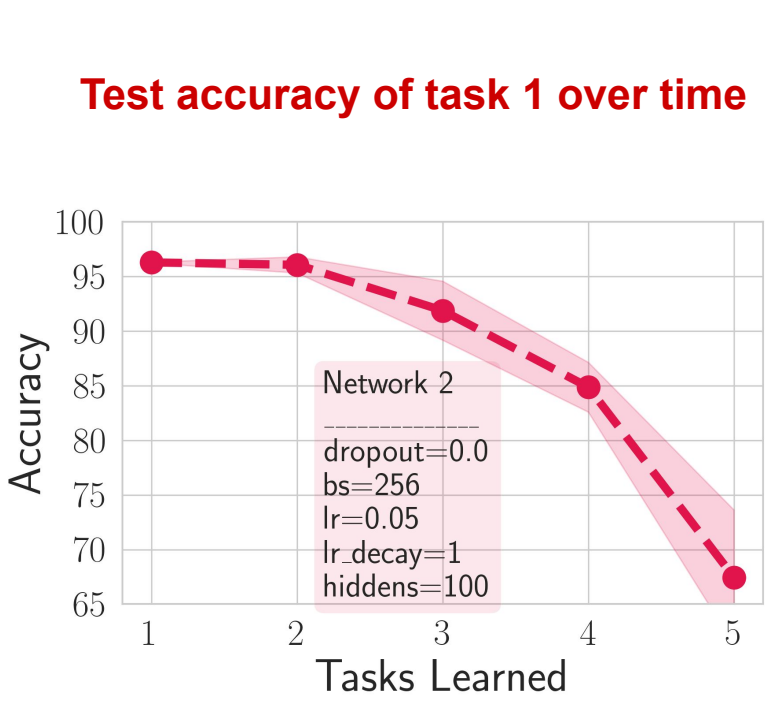
Task 1  0

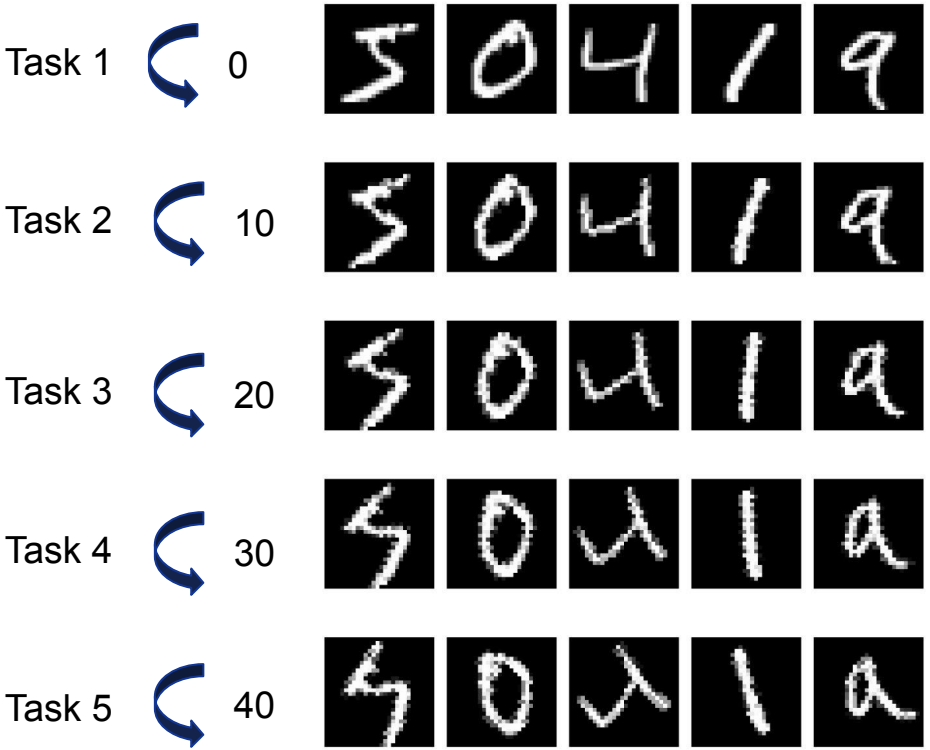
Task 5  40

Is forgetting really catastrophic?

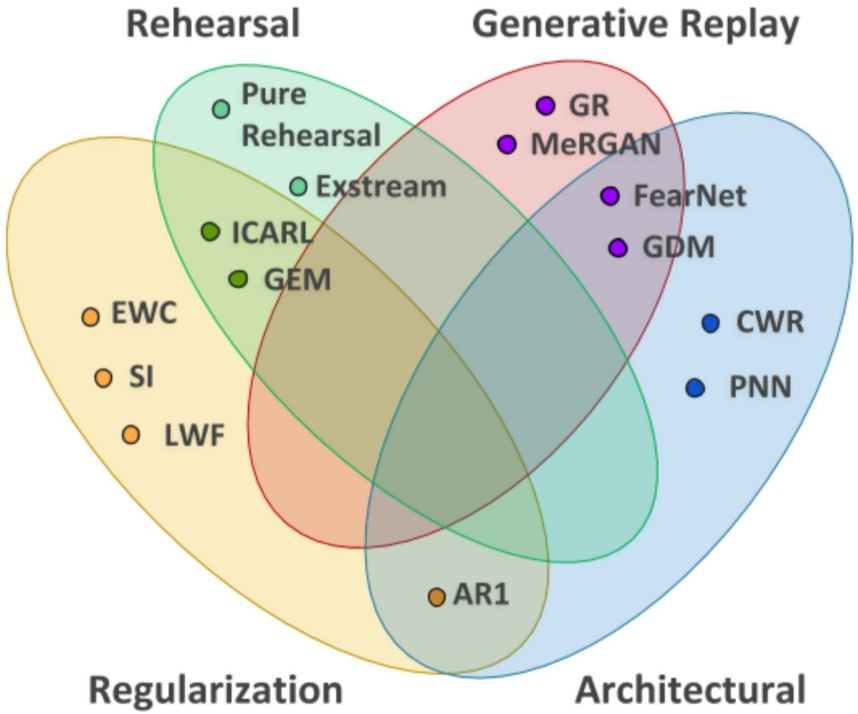


Is forgetting really catastrophic?



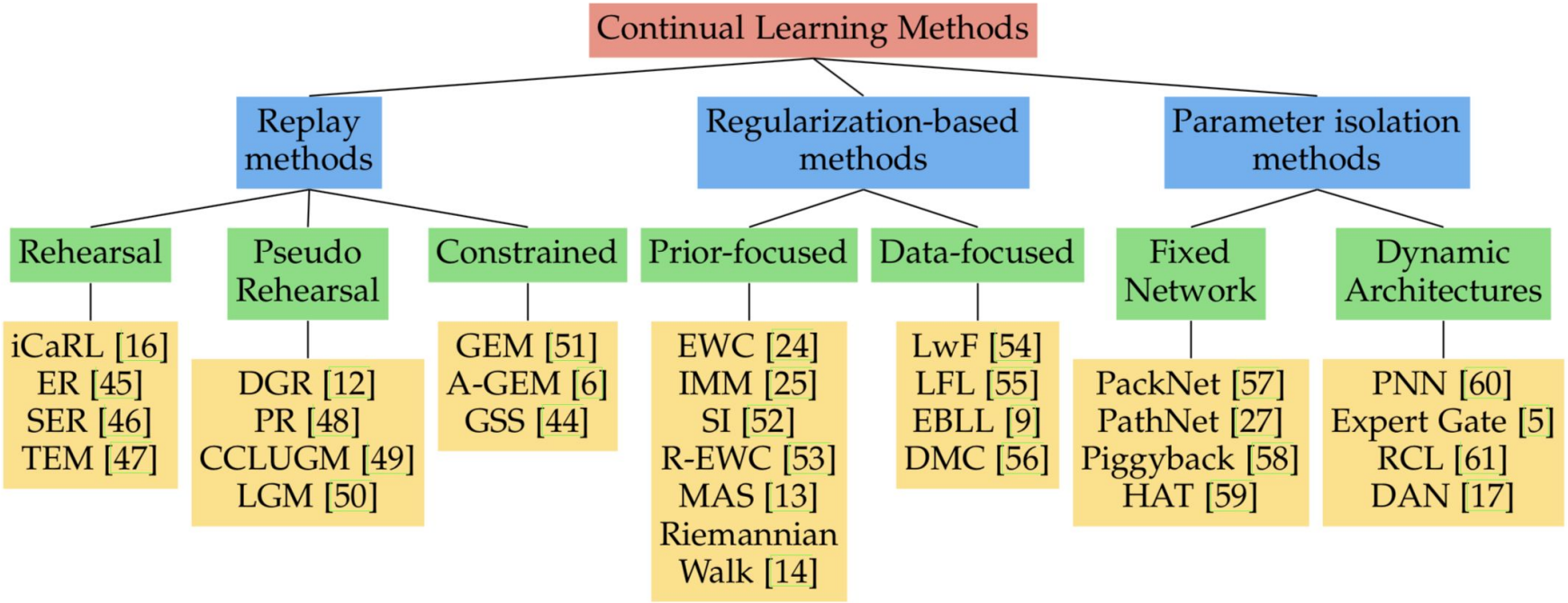


Different solutions



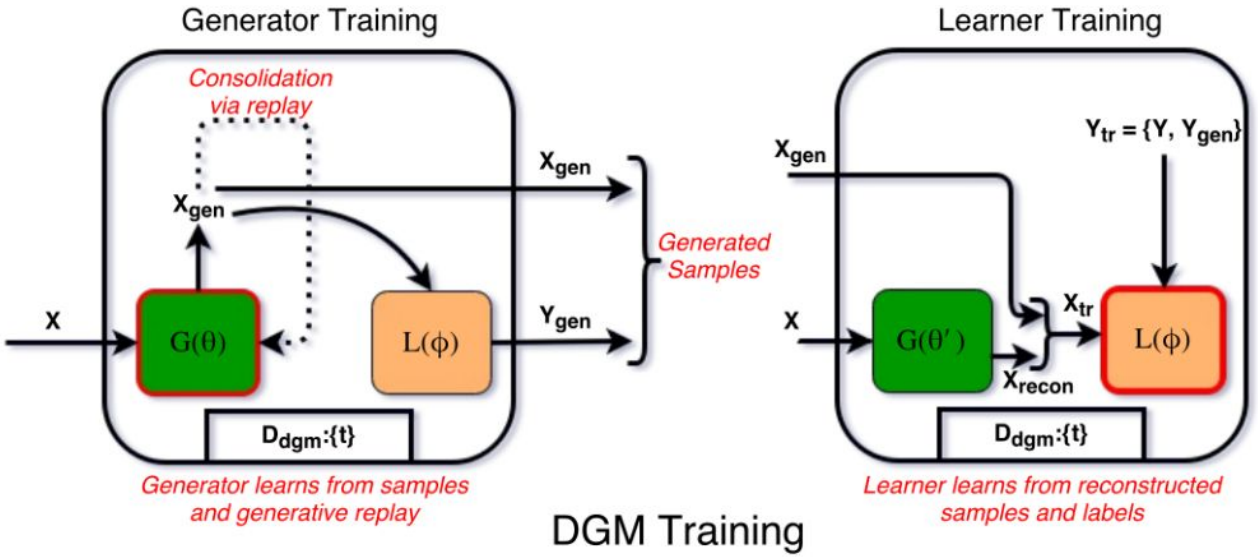
Continual Learning for Robotics: Definition, Framework, Learning Strategies, Opportunities and Challenges, Lesort et al., 2019

Different solutions



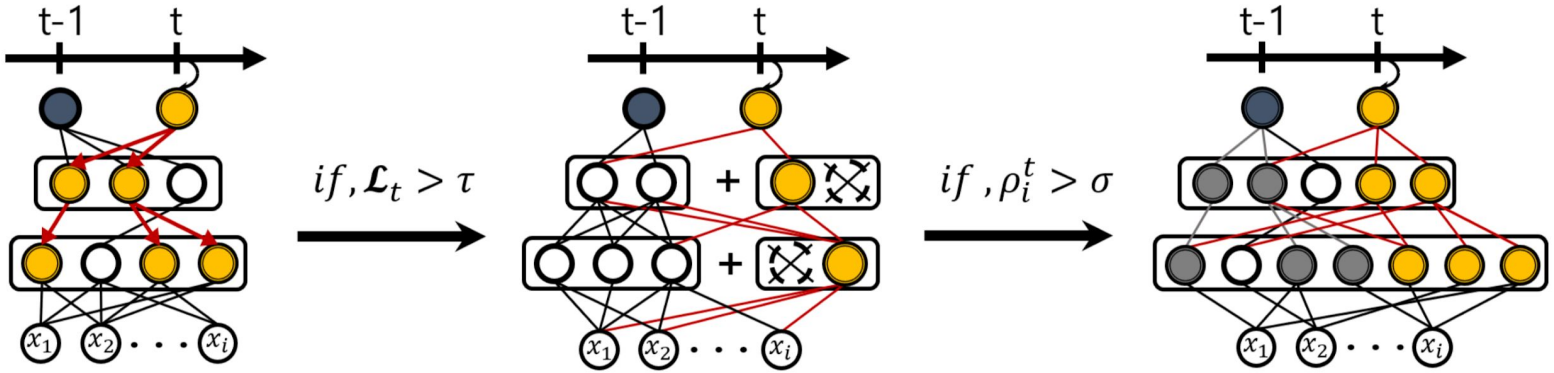
A continual learning survey: Defying forgetting in classification tasks, Lange et al., 2020

Repetition based



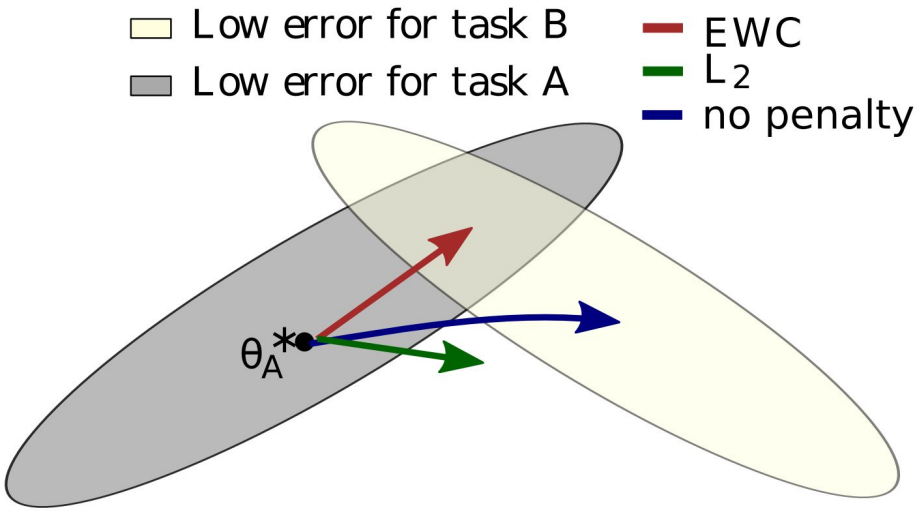
Deep Generative Dual Memory Network for Continual Learning, Kamra et al., ICLR 2018

Expansion based



Lifelong Learning with Dynamically Expandable Networks, Yoon et al., 2019

Regularization based



Overcoming catastrophic forgetting in neural networks, Kirkpatrick et al., 2016

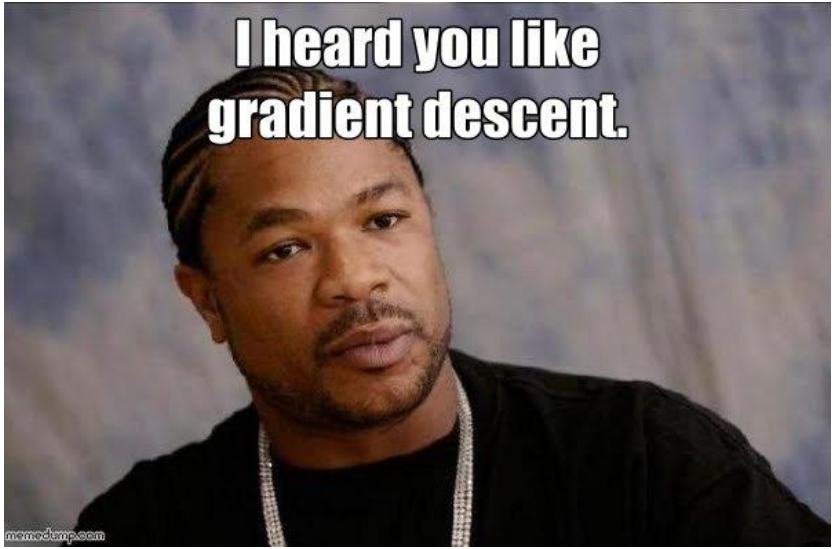
Orthogonal Gradient Descent for Continual Learning

A regularization perspective

Farajtabar, Azizan, Mott, Li, https://arxiv.org/abs/1910.07104, AISTATS 2020

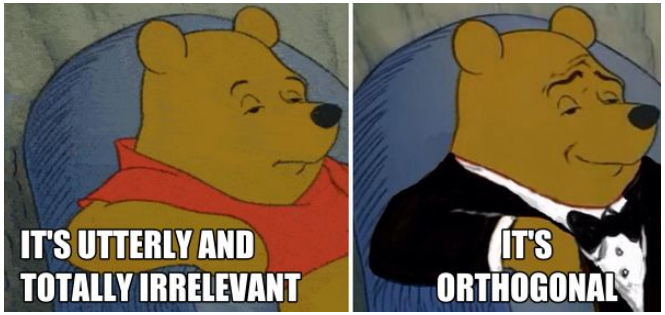
Whose guilt is this? stochastic gradient descent?

- The optimizers produce gradients that are oblivious to past knowledge
- By design are purely a function of the current minibatch
- Desirable when the training data is iid, but is not desirable when the training distribution shifts over time.

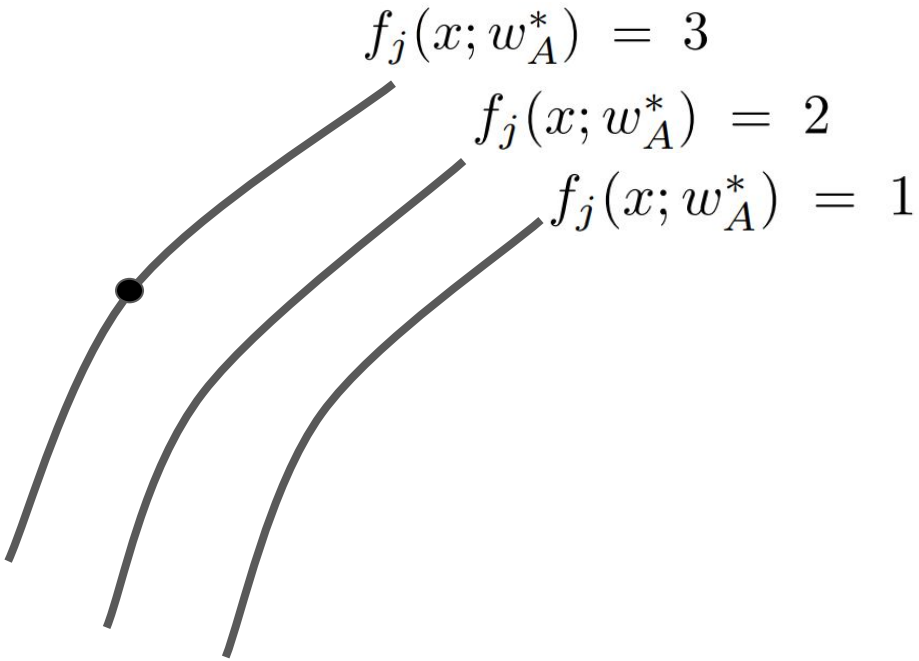


Our solution!

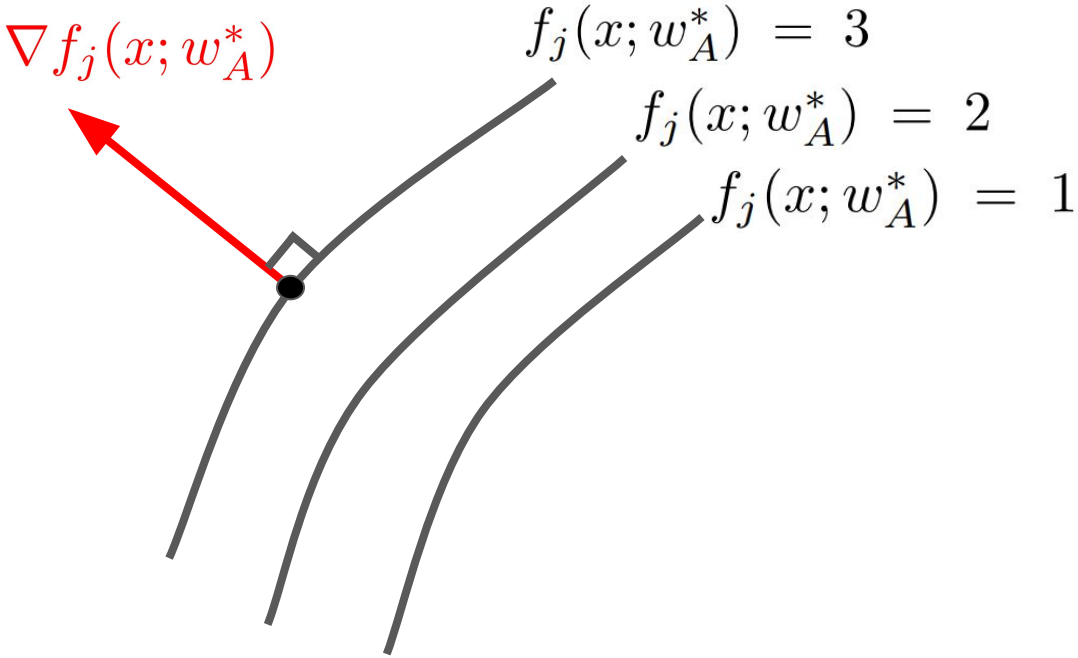
- Orthogonal Gradient Descent (OGD)
- Preserve the previously acquired knowledge by maintaining a space consisting of the gradient directions of the neural network predictions on previous ones
- Any update orthogonal to this gradient space change the output of the network minimally
- Project the loss gradients of new samples perpendicular to this gradient space before applying back-propagation



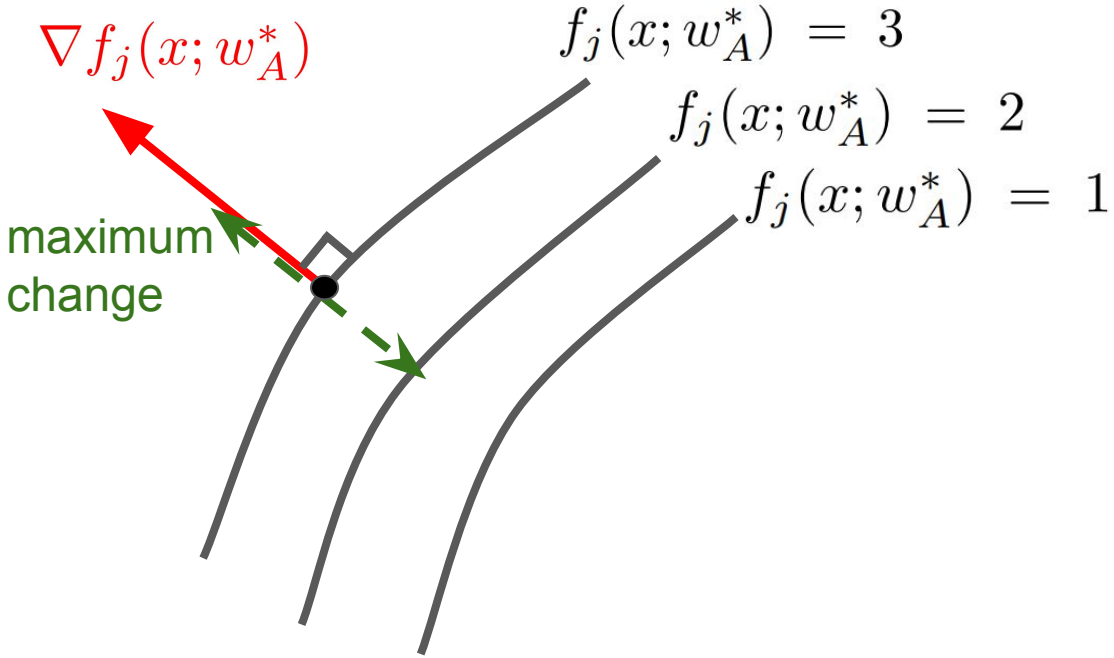
How the network predictions can change minimally?



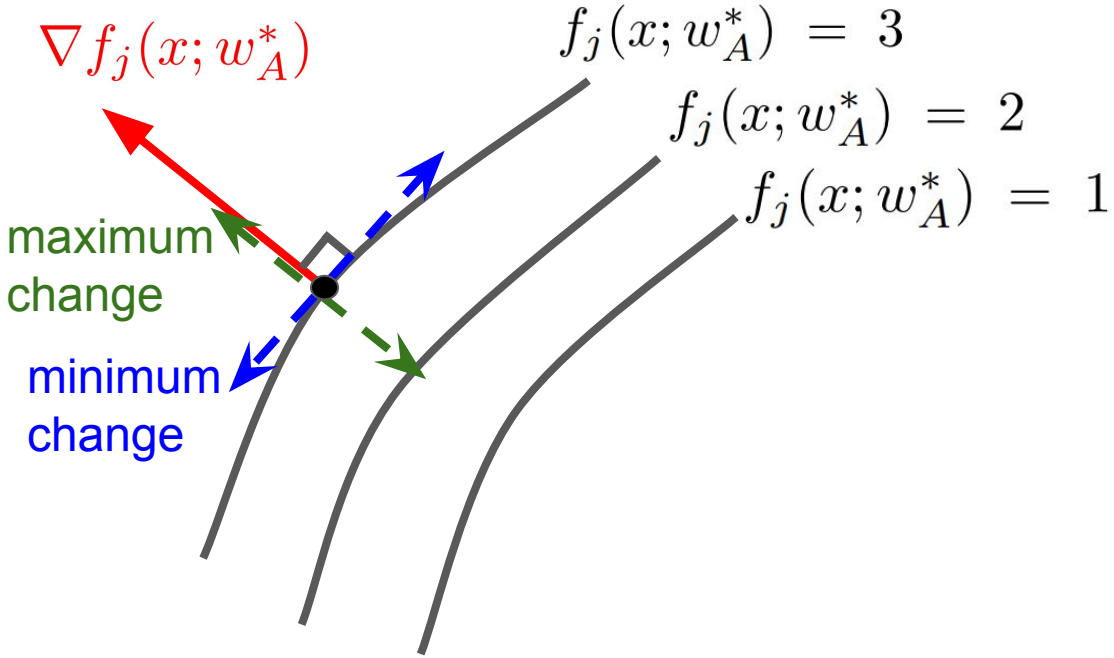
How the network predictions can change minimally?



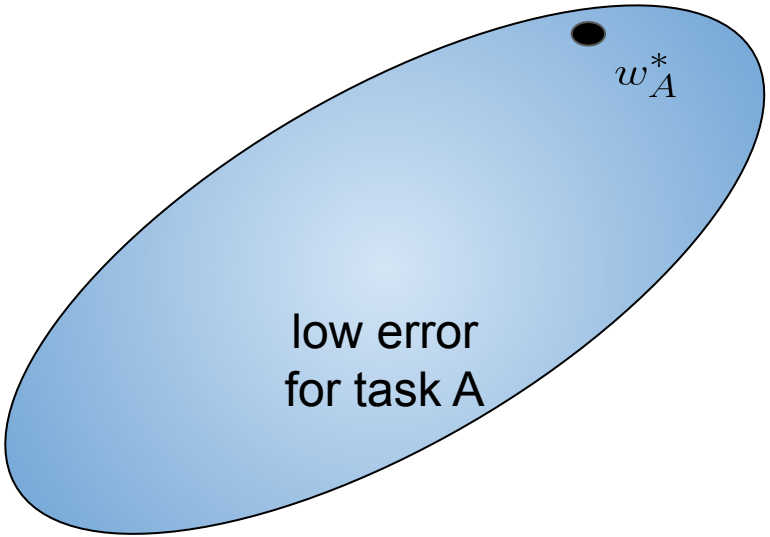
How the network predictions can change minimally?



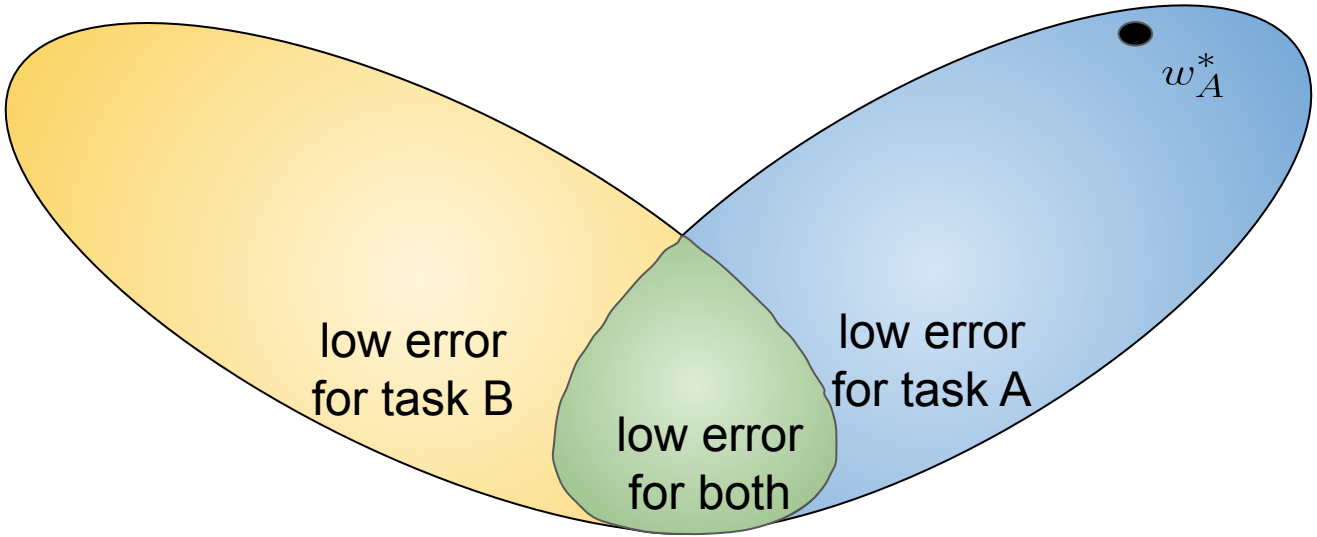
How the network predictions can change minimally?



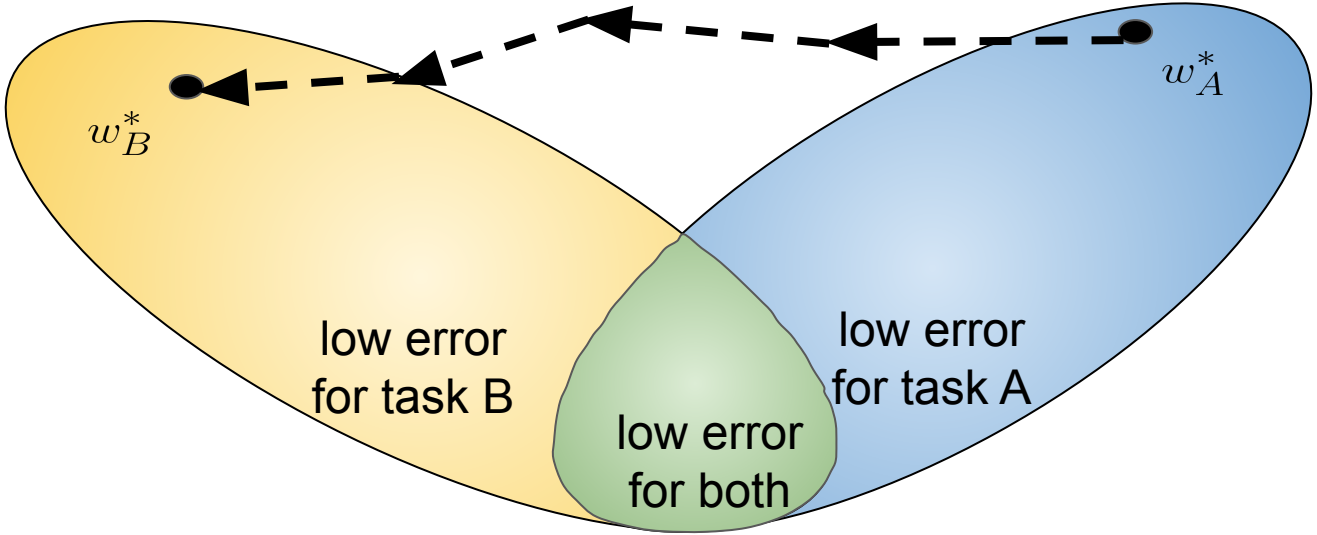
OGD illustration



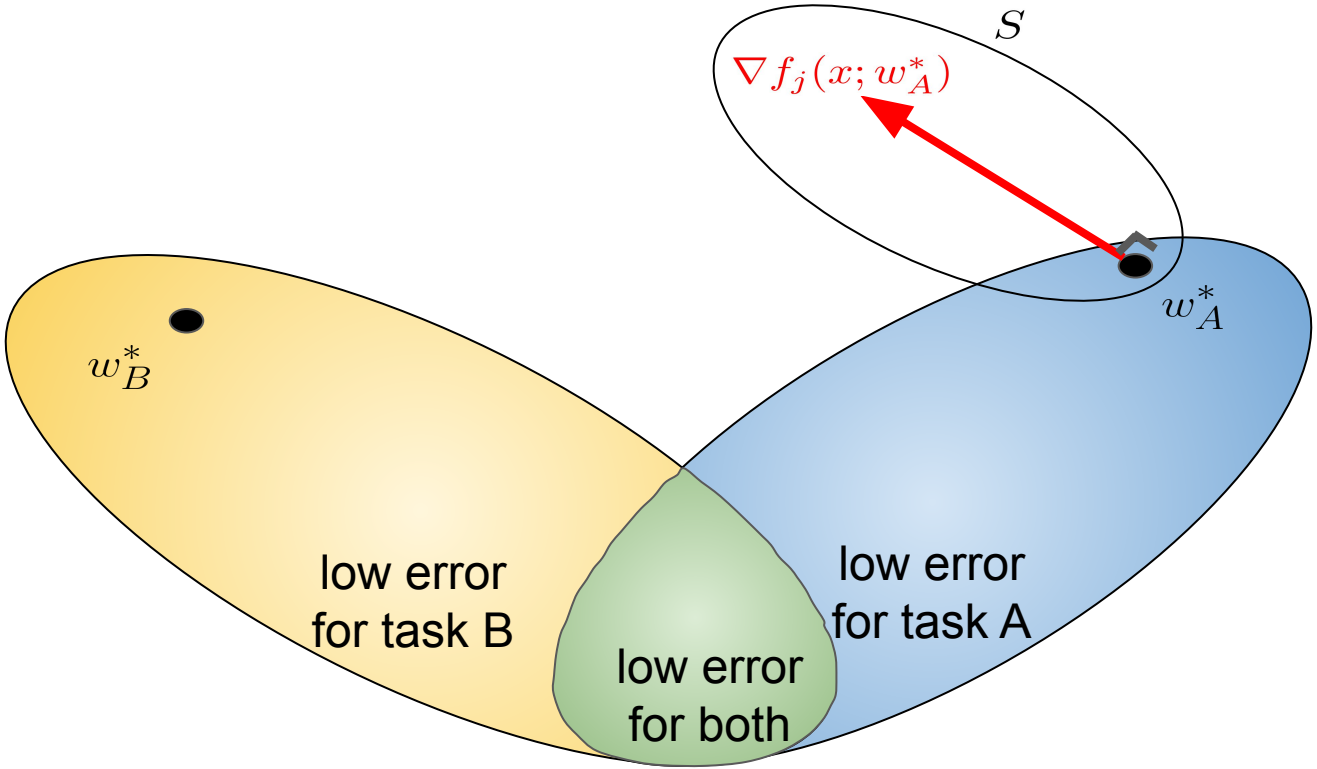
OGD illustration



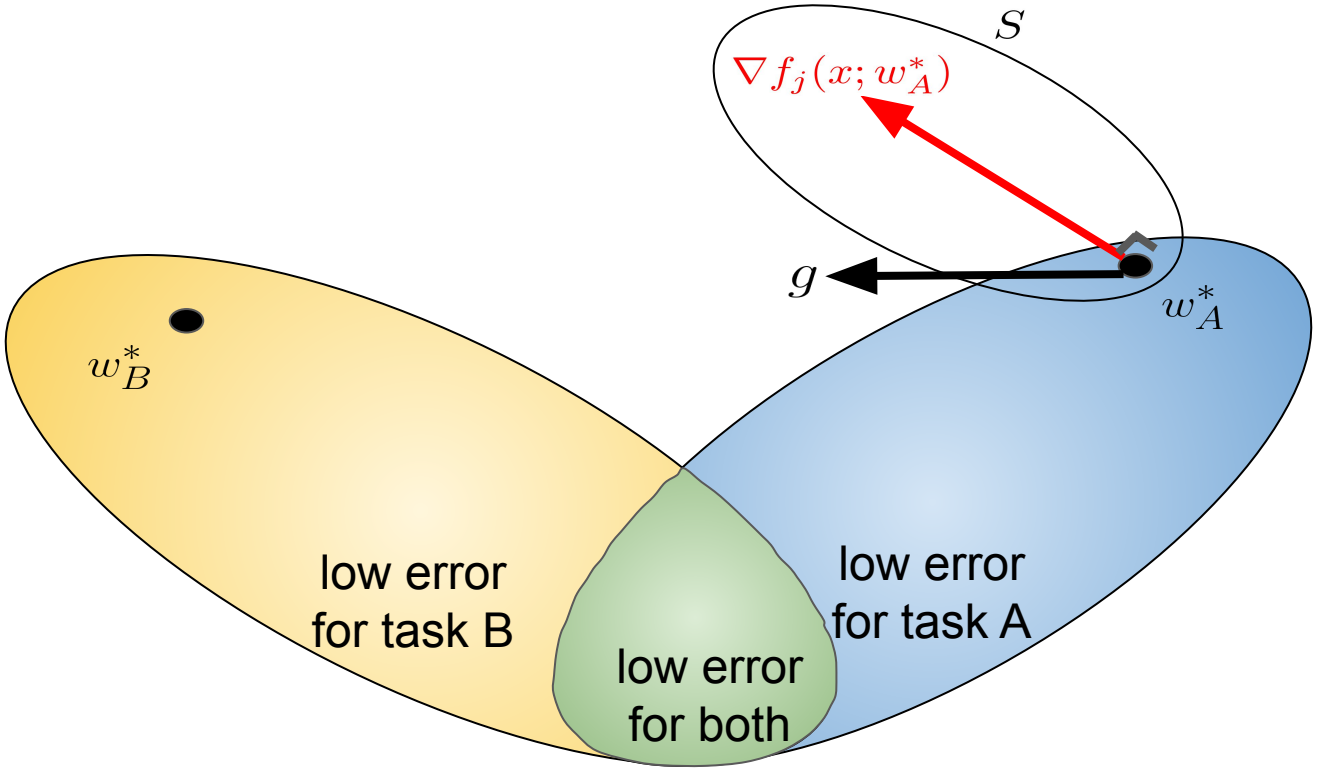
OGD illustration



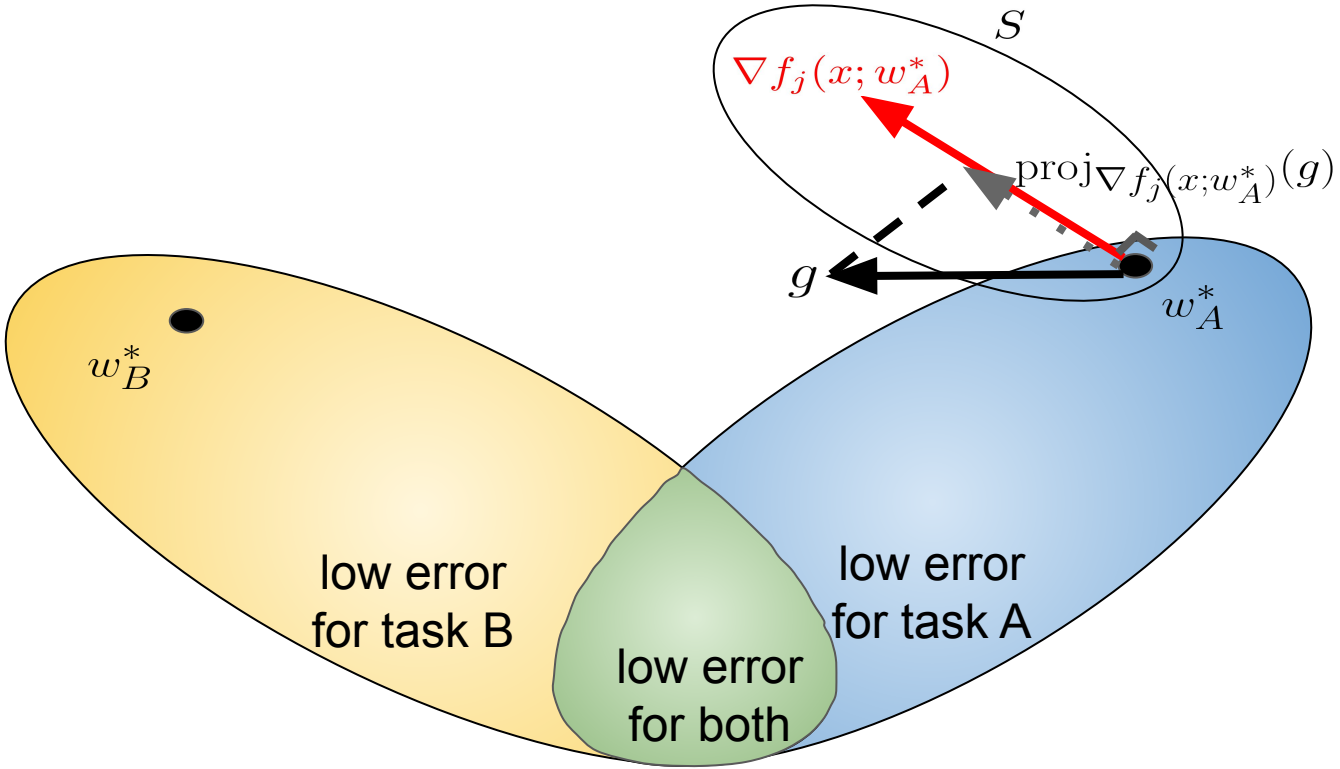
OGD illustration



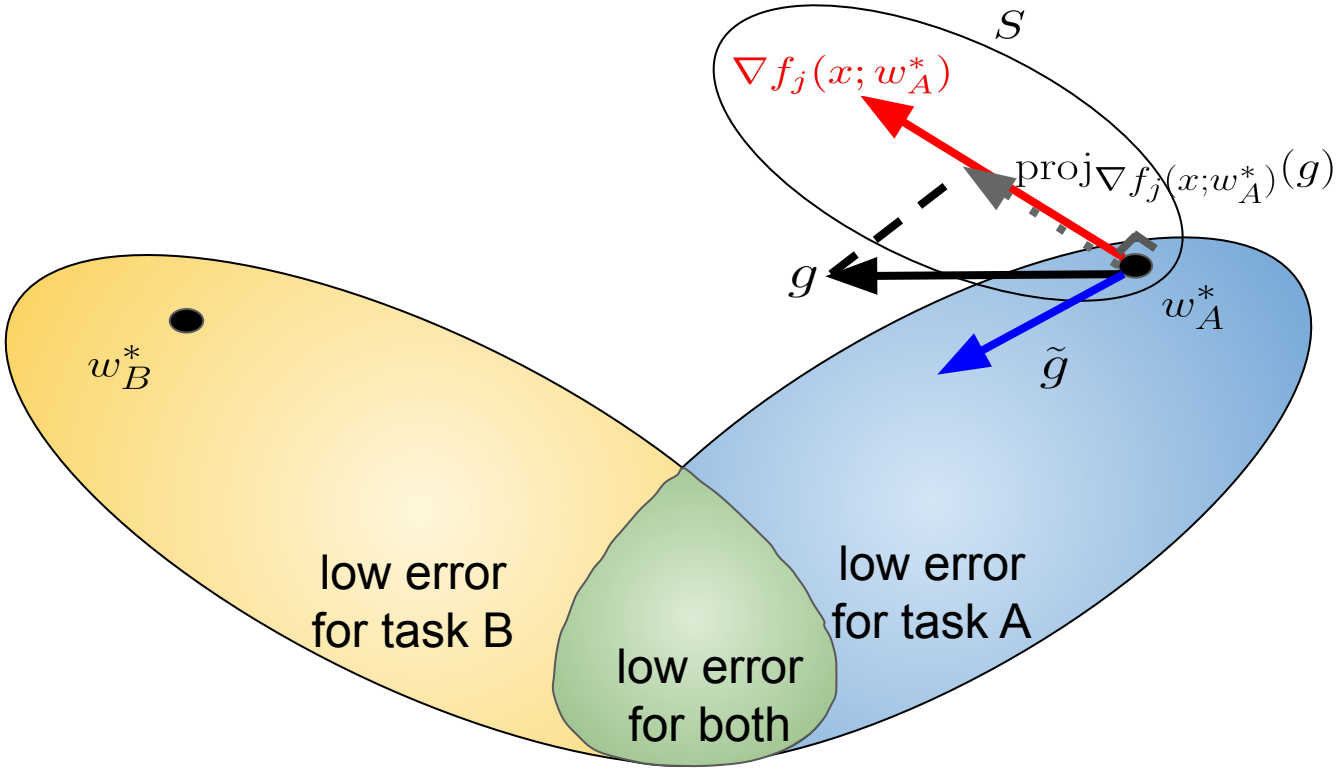
OGD illustration



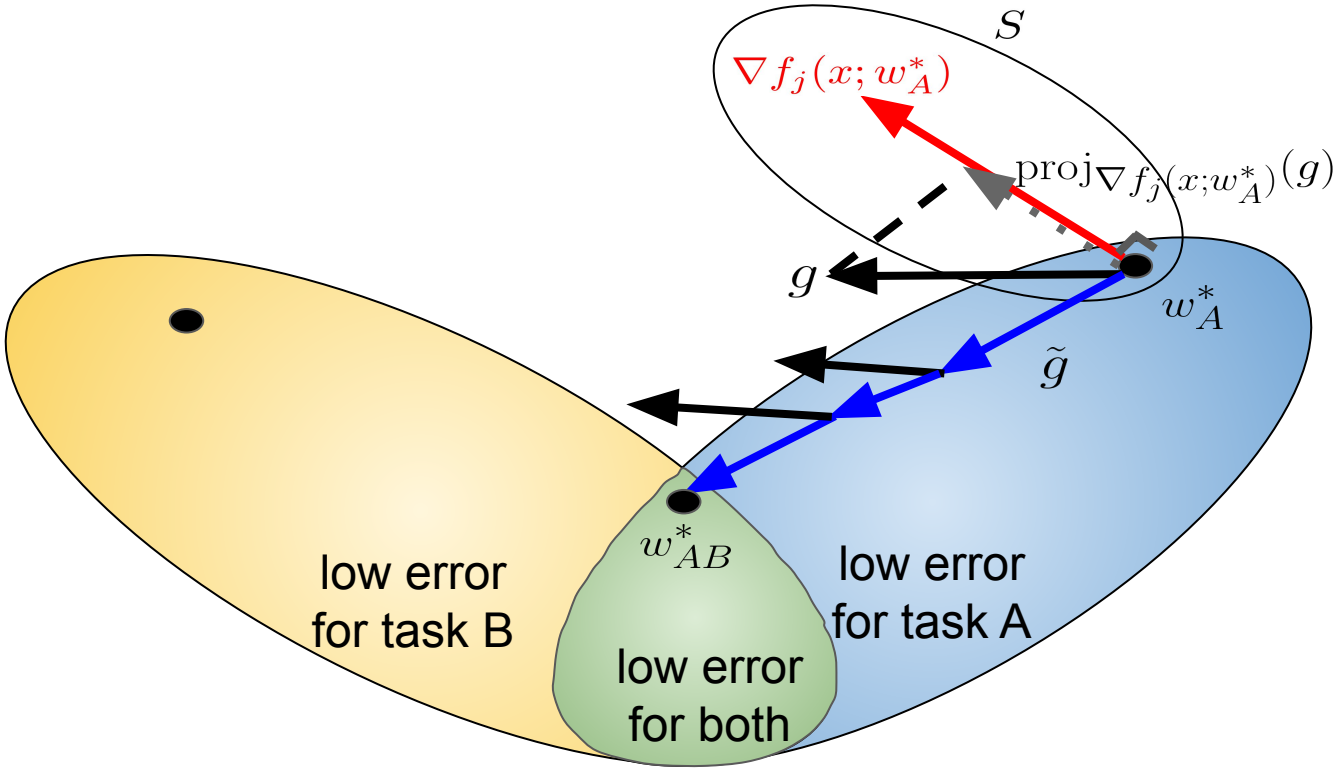
OGD illustration



OGD illustration



OGD illustration



Experiments

- Network: Relu network with 2 hidden layer with 100 units
- Storage size: 200 gradients
- Learning rate: 0.001
- 5 epochs of training per task

Experiments

- Network: Relu network with 2 hidden layer with 100 units
- Storage size: 200 gradients
- Learning rate: 0.001
- 5 epochs of training per task
- Baselines:
    - EWC: one of the pioneering regularization based methods that uses fisher information diagonals as importance weights.
    - A-GEM: using loss gradients of stored previous data in an inequality constrained optimization.
    - SGD: vanilla Stochastic Gradient Descent
    - MTL: Multi Task Learning baseline with full access to previous data.

Permuted MNIST

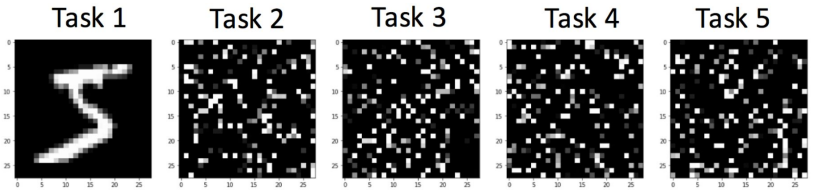
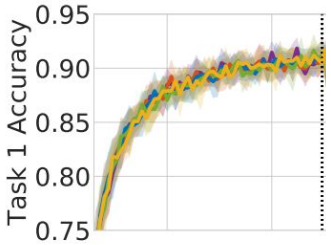
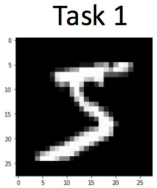


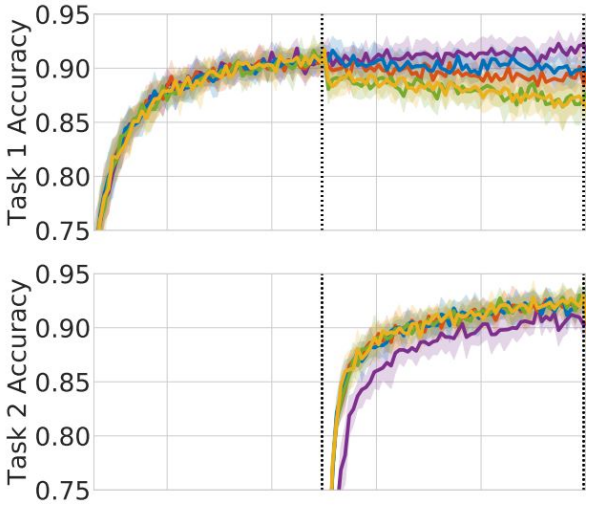
Image from: Three scenarios for continual learning

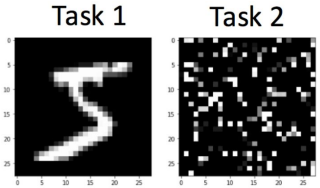
Permuted MNIST



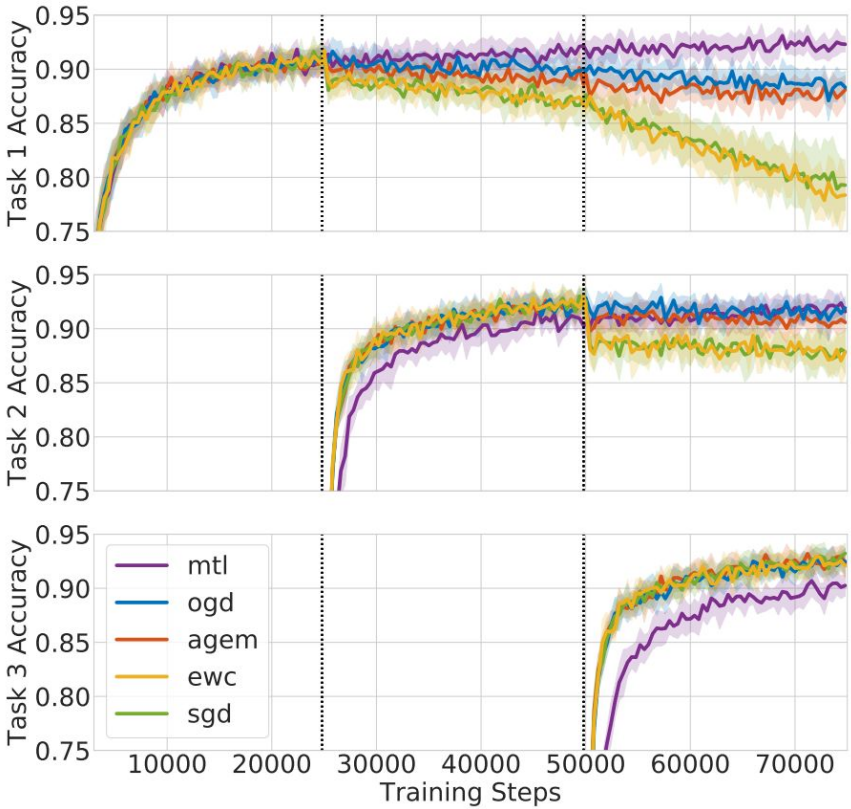


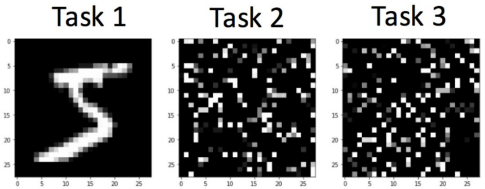
Permuted MNIST





Permuted MNIST





Permuted MNIST

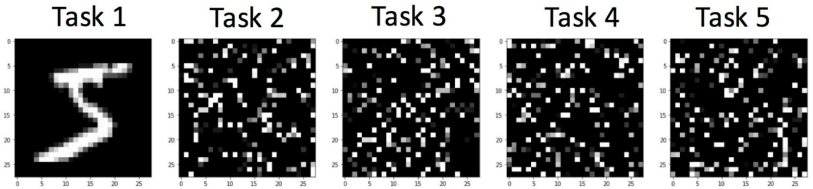
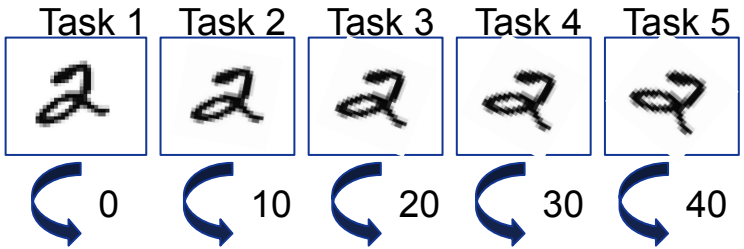


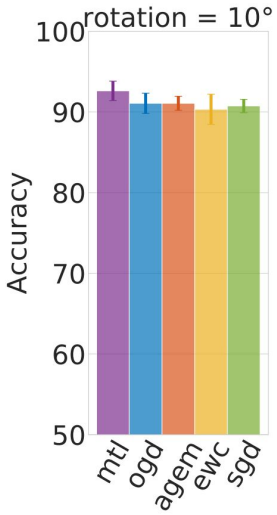
Image from: Three scenarios for continual learning

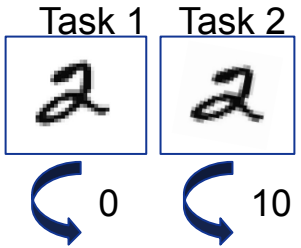
	Accuracy ± Std. (%)				
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**
MTL	93.2 ± 1.3	91.5 ± 0.5	91.3 ± 0.7	91.3 ± 0.6	88.4 ± 0.8
OGD	79.5 ± 2.3	$\mathbf{88.9} \pm 0.7$	$\mathbf{89.6} \pm 0.3$	$\mathbf{91.8} \pm 0.9$	92.4 ± 1.1
A-GEM	$\mathbf{85.5} \pm 1.7$	87.0 ± 1.5	$\mathbf{89.6} \pm 1.1$	91.2 ± 0.8	$\mathbf{93.9} \pm 1.0$
EWC	64.5 ± 2.9	77.1 ± 2.3	80.4 ± 2.1	87.9 ± 1.3	93.0 ± 0.5
SGD	60.6 ± 4.3	77.6 ± 1.4	79.9 ± 2.1	87.7 ± 2.9	92.4 ± 1.1

Rotated MNIST

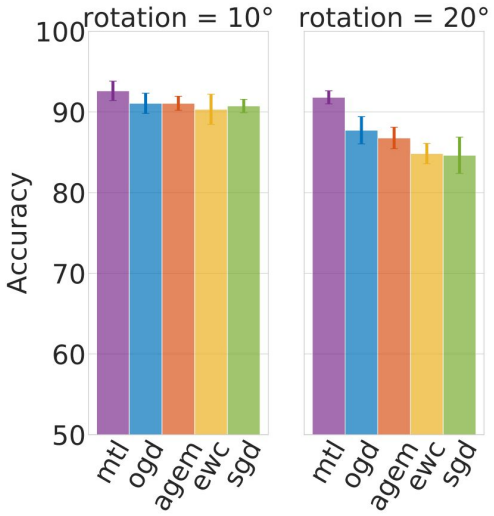


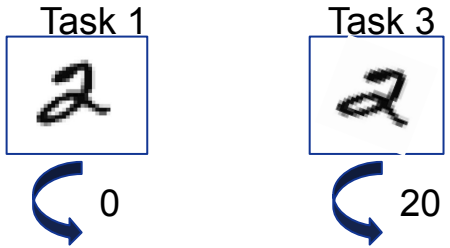
Rotated MNIST



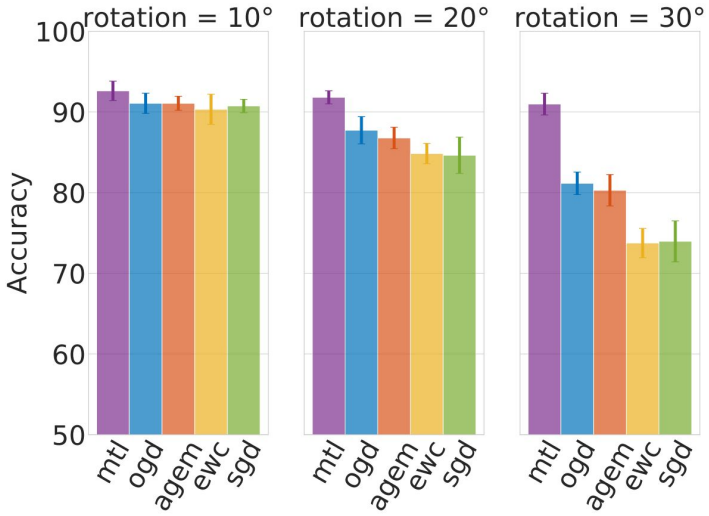


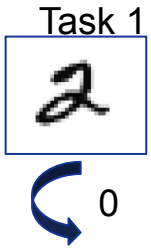
Rotated MNIST

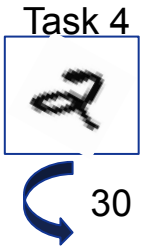




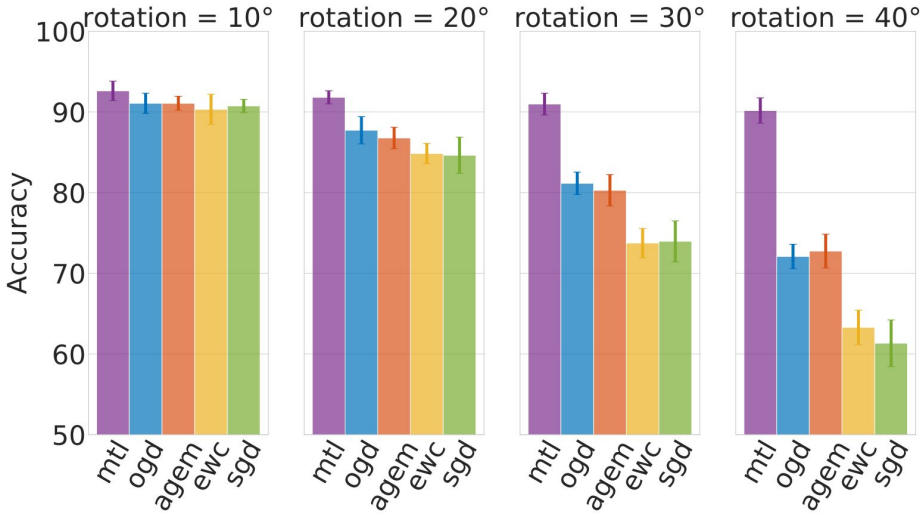
Rotated MNIST

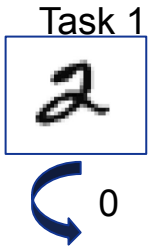


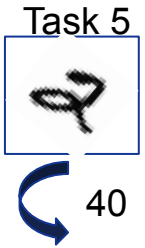




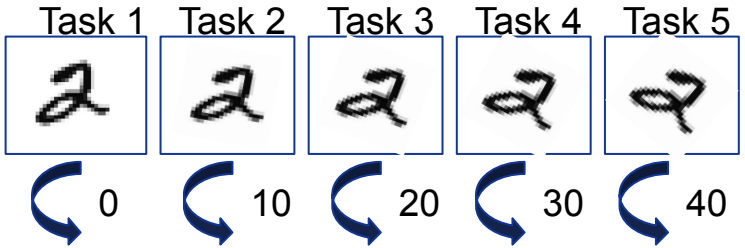
Rotated MNIST







Rotated MNIST



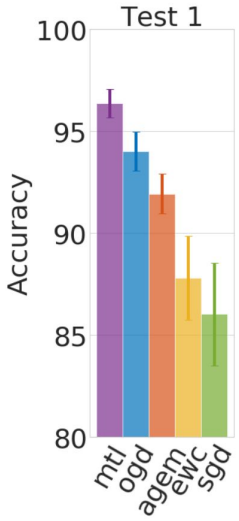
	Accuracy ± Std. (%)				
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**
MTL	92.1 ± 0.9	94.3 ± 0.9	95.2 ± 0.9	93.4 ± 1.1	90.5 ± 1.5
OGD	$\mathbf{75.6} \pm 2.1$	$\mathbf{86.6} \pm 1.3$	$\mathbf{91.7} \pm 1.1$	94.3 ± 0.8	93.4 ± 1.1
A-GEM	72.6 ± 1.8	84.4 ± 1.6	91.0 ± 1.1	93.9 ± 0.6	$\mathbf{94.6} \pm 1.1$
EWC	61.9 ± 2.0	78.1 ± 1.8	89.0 ± 1.6	94.4 ± 0.7	93.9 ± 0.6
SGD	62.9 ± 1.0	76.5 ± 1.5	88.6 ± 1.4	$\mathbf{95.1} \pm 0.5$	94.1 ± 1.1

Split MNIST

Task 1

Task 2

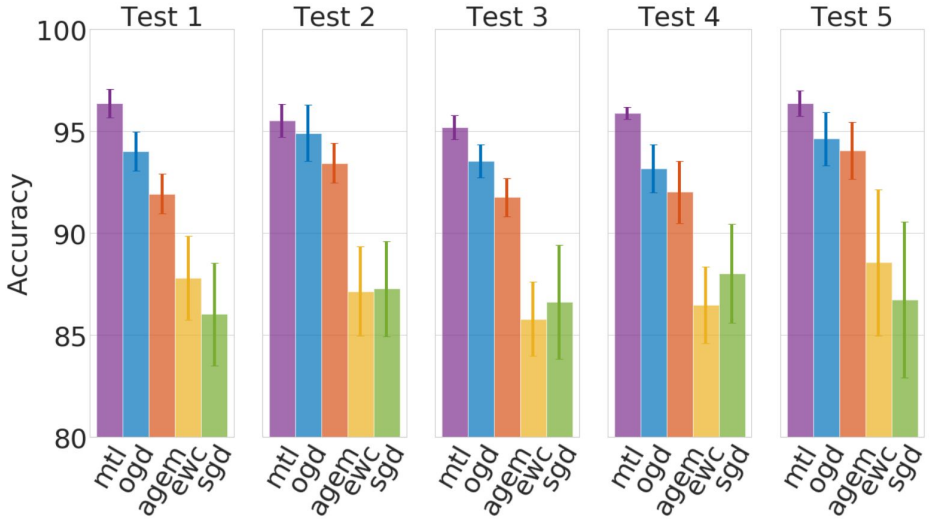
Split MNIST



Task 1

Task 2

Split MNIST



Task 1

Task 2

Split MNIST

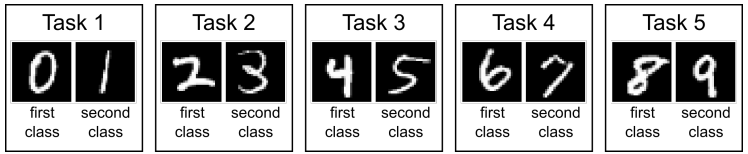


Image from: Three scenarios for continual learning

	Accuracy ± Std. (%)				
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**
MTL	99.6 ± 0.2	99.8 ± 0.1	98.8 ± 0.2	98.2 ± 0.4	99.1 ± 0.2
OGD	$\mathbf{98.6} \pm 0.8$	$\mathbf{99.5} \pm 0.1$	$\mathbf{98.0} \pm 0.5$	$\mathbf{98.8} \pm 0.5$	99.2 ± 0.3
A-GEM	92.9 ± 2.6	96.3 ± 2.1	86.5 ± 1.6	92.3 ± 2.3	99.3 ± 0.2
EWC	90.2 ± 5.7	98.9 ± 0.2	91.1 ± 3.5	94.4 ± 2.0	99.3 ± 0.2
SGD	88.2 ± 5.9	98.4 ± 0.9	90.3 ± 4.5	95.2 ± 1.0	$\mathbf{99.4} \pm 0.2$

Conclusion and future works

Conclusion and future works

- Accuracy:
  - Still some forgetting happens:
  - Higher order invariant spaces

Conclusion and future works

- Accuracy:
  - Still some forgetting happens:
  - Higher order invariant spaces
- Scalability:
  - Storage scales proportional to number of tasks
  - Preserve significant and important directions

Conclusion and future works

- Accuracy:
  - Still some forgetting happens:
  - Higher order invariant spaces
- Scalability:
  - Storage scales proportional to number of tasks
  - Preserve significant and important directions
- Task diversity:
  - Failure when tasks are very dissimilar

Conclusion and future works

- Accuracy:
  - Still some forgetting happens:
  - Higher order invariant spaces
- Scalability:
  - Storage scales proportional to number of tasks
  - Preserve significant and important directions
- Task diversity:
  - Failure when tasks are very dissimilar
- Hyperparameters:
  - Sensitivity to learning rate → meta learning of the learning rate

Conclusion and future works

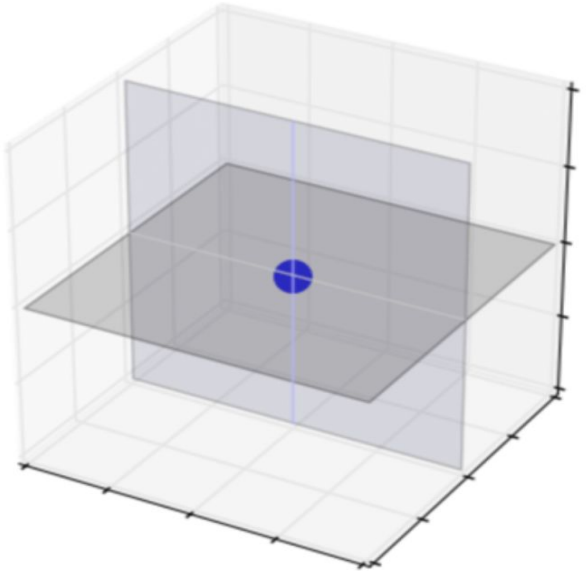
- Other optimizers
  - Adam, Adagrad, etc

Conclusion and future works

- Other optimizers
  - Adam, Adagrad, etc
- Beyond Continual Learning!
  - Whenever you want to have minimal interferences with the previously learned data

Conclusion and future works

- Other optimizers
  - Adam, Adagrad, etc
- Beyond Continual Learning!
  - Whenever you want to have minimal interferences with the previously learned data
- Orthogonal in some intermediate representation not in input data. Representation Learning for OGD



Dropout as an Implicit Gating Mechanism for Continual Learning

A pathway perspective

Mirzadeh, Farajtabar, Ghasemzadeh https://arxiv.org/abs/2004.11545, Continual Learning workshop at CVPR 2020

Origin of the work

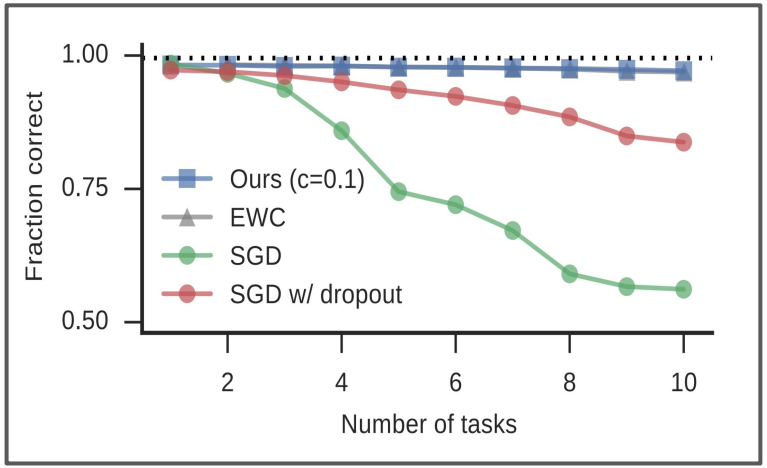
We find that in most cases, dropout increases the optimal size of the net, so the resistance to forgetting may be explained mostly by the larger nets having greater capacity. However, this effect is not consistent, and when using dissimilar task pairs, dropout usually decreases the size of the net. **This suggests dropout may have other more subtle beneficial effects to characterize in the future.**

“Goodfellow et al., 2013”

Origin of the work

We find that in most cases, dropout increases the optimal size of the net, so the resistance to forgetting may be explained mostly by the larger nets having greater capacity. However, this effect is not consistent, and when using dissimilar task pairs, dropout usually decreases the size of the net. **This suggests dropout may have other more subtle beneficial effects to characterize in the future.**

"Goodfellow et al., 2013"

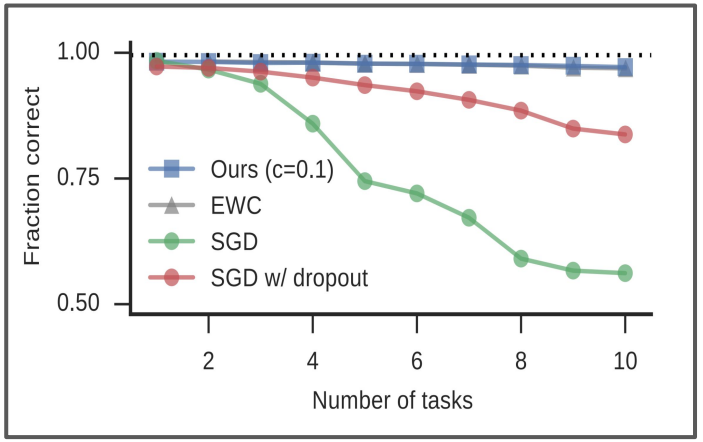


Zenke et al., 2017

Origin of the work

We find that it is always best to train using the **dropout** algorithm– the dropout algorithm is consistently best at adapting to the new task, remembering the old task, and has the best tradeoff curve between these two extremes

"Goodfellow et al., 2013"



Zenke et al., 2017

Question:
WHY does dropout help in continual learning?

March 2020

Origin of the work

Question:
WHY does dropout help In continual learning?

Answer:
Maybe because of the implicit gating mechanism and regularization

March 2020

April 2020

Implicit gating

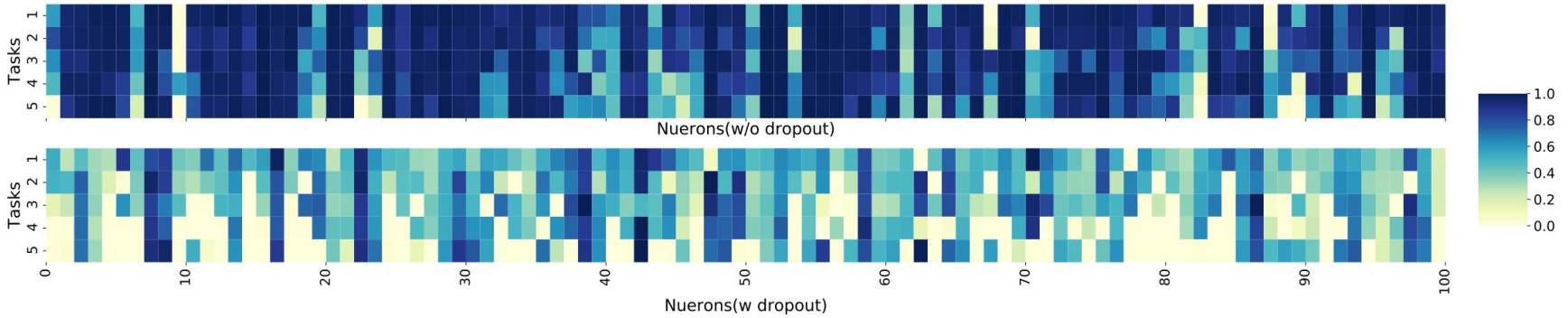


Figure: The effect of dropout on the activation(firing) pattern of neurons

Implicit gating

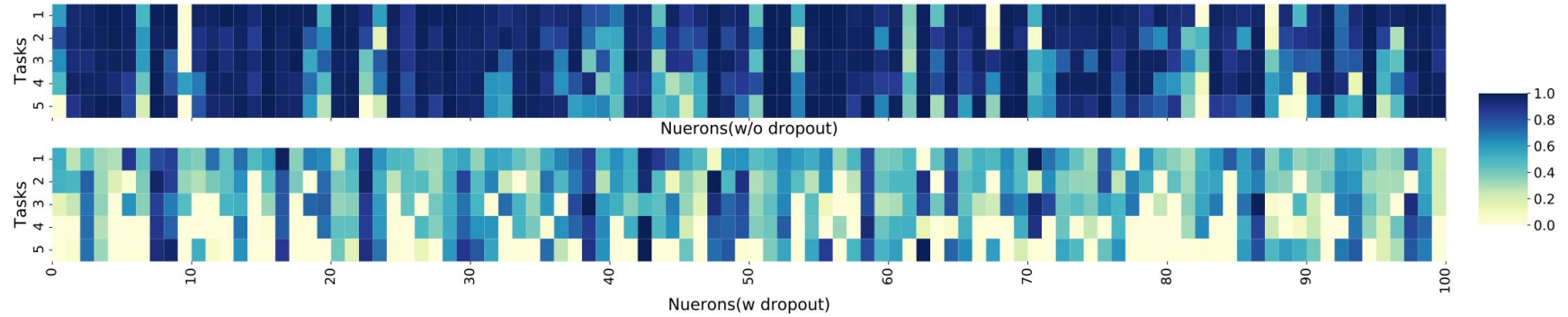
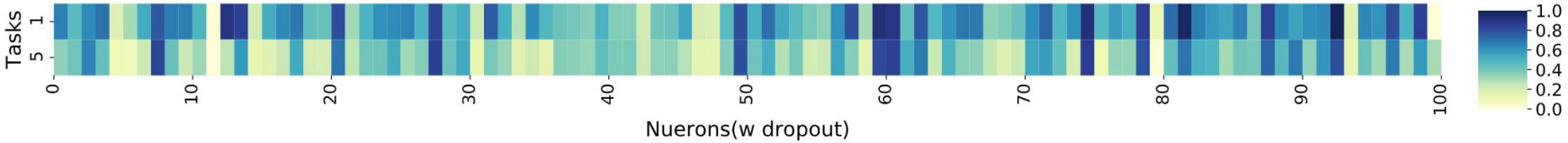


Figure: The effect of dropout on the activation(firing) pattern of neurons



Increasing dropout rate to gain stability

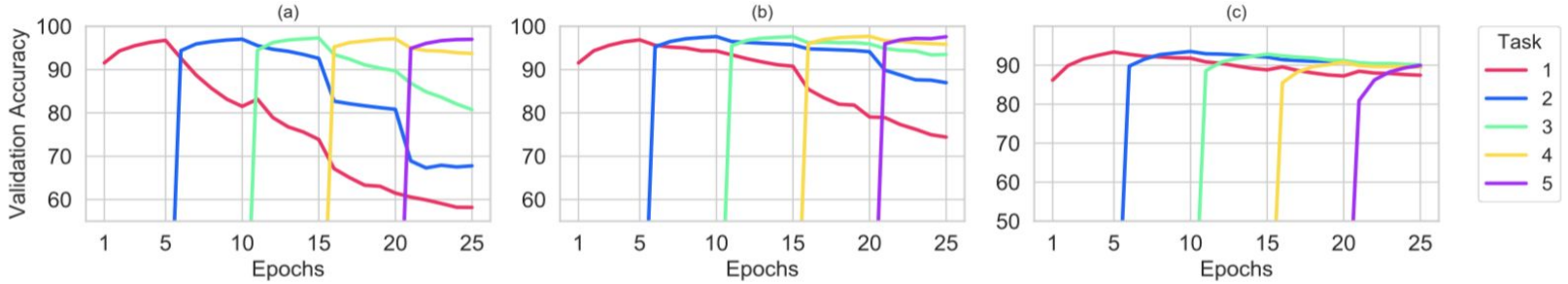


Figure 2. *Permuted MNIST*- Increasing the stability and reducing the plasticity from left to right by increasing the the dropout rate and learning rate decay.

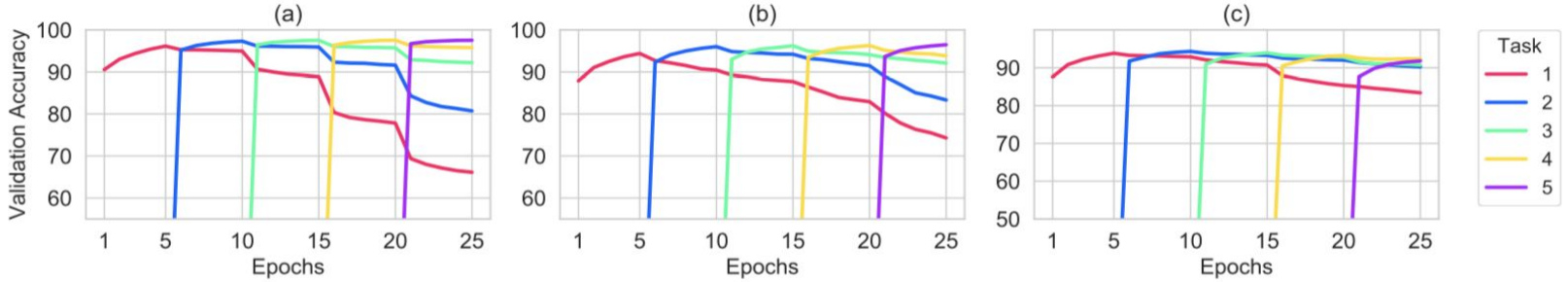


Figure 3. *Rotated MNIST*- Increasing the stability and reducing the plasticity from left to right by increasing the the dropout rate and learning rate decay.

Dropout training of Rotated MNIST

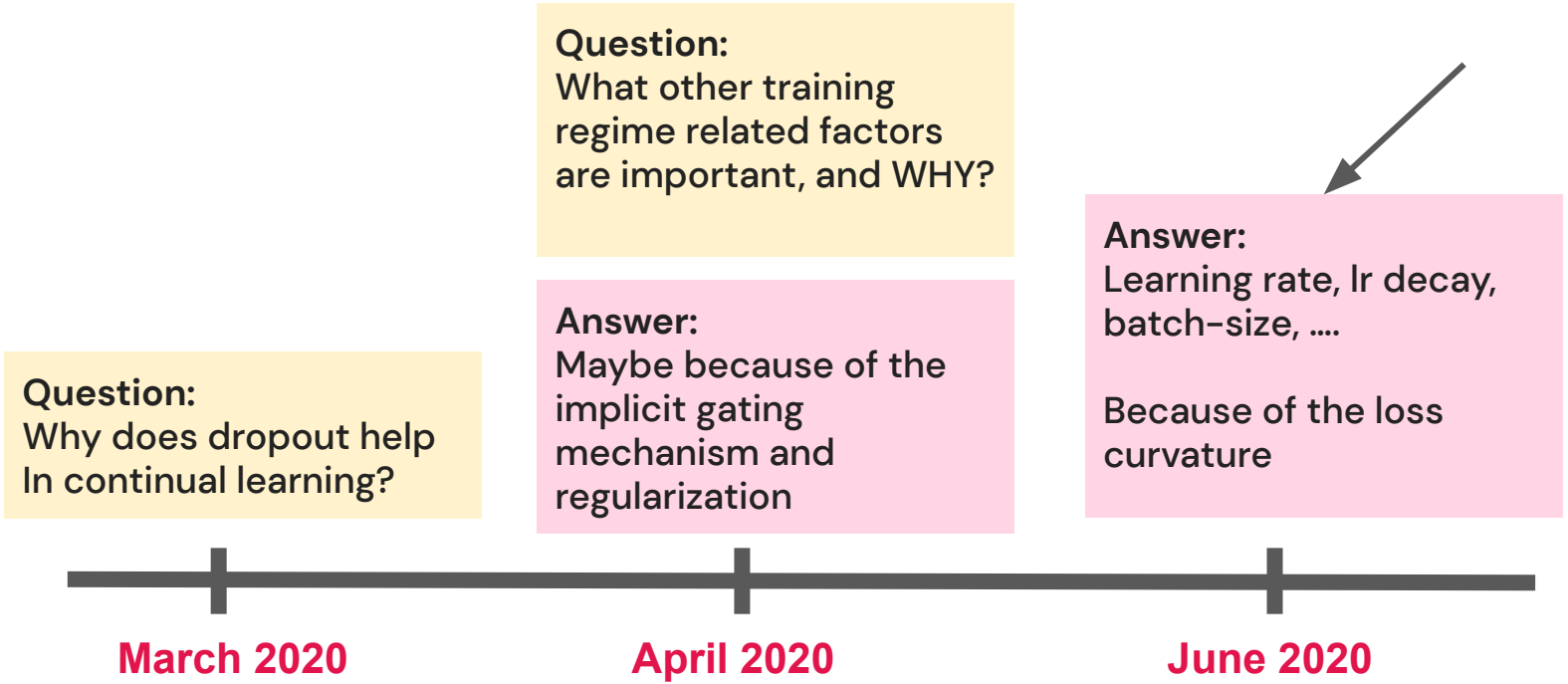
	Accuracy ± std (%)				
	Task 1	**Task 2**	**Task 3**	**Task 4**	**Task 5**
MTL	92.1 ± 0.9	94.3 ± 0.9	95.2 ± 0.9	93.4 ± 1.1	90.5 ± 1.5
OGD	75.6 ± 2.1	86.6 ± 1.3	91.7 ± 1.1	94.3 ± 0.8	93.4 ± 1.1
A-GEM	72.6 ± 1.8	84.4 ± 1.6	91.0 ± 1.1	93.9 ± 0.6	**94.6 ± 1.0**
EWC	67.9 ± 2.0	78.1 ± 1.8	89.0 ± 1.6	94.4 ± 0.7	93.9 ± 0.6
SGD	65.9 ± 1.8	77.5 ± 1.5	88.6 ± 1.4	**95.1 ± 0.5**	94.1 ± 1.1
SGD+Dropout	**81.1 ± 1.1**	**89.3 ± 2.4**	**92.1 ± 2.2**	93.4 ± 1.8	92.8 ± 0.5

Table 2. *Rotated MNIST*: The validation accuracy of the model for each task, after being trained on all tasks in sequence.

Conclusion and future works

- Sparsity in natural intelligence
  - Neural activities and their connectivity are sparse
  - Neurons are activated via many sparse patterns
- Smarter gating
  - Manually selecting pathways
  - Generalized dropout
  - [Stochastic] Block membership or cluster priors over activations
  - Dirichlet process for adapting to number of tasks on the fly
- Recurrent models and attention modeling

Conclusion


Question:
What other training regime related factors are important, and WHY?
Answer:
Learning rate, lr decay, batch-size, ...
Because of the loss curvature
Answer:
Maybe because of the implicit gating mechanism and regularization
Question:
Why does dropout help In continual learning?
March 2020
April 2020
June 2020

Understanding the Role of Training Regimes in Continual Learning

A wideness/sharpness perspective of loss landscape

Mirzadeh, Farajtabar, Ghasemzadeh, Pascanu, https://arxiv.org/abs/2006.06958, Continual Learning workshop at ICML 2020

Research questions

Q1: How important a training regime is in continual learning?

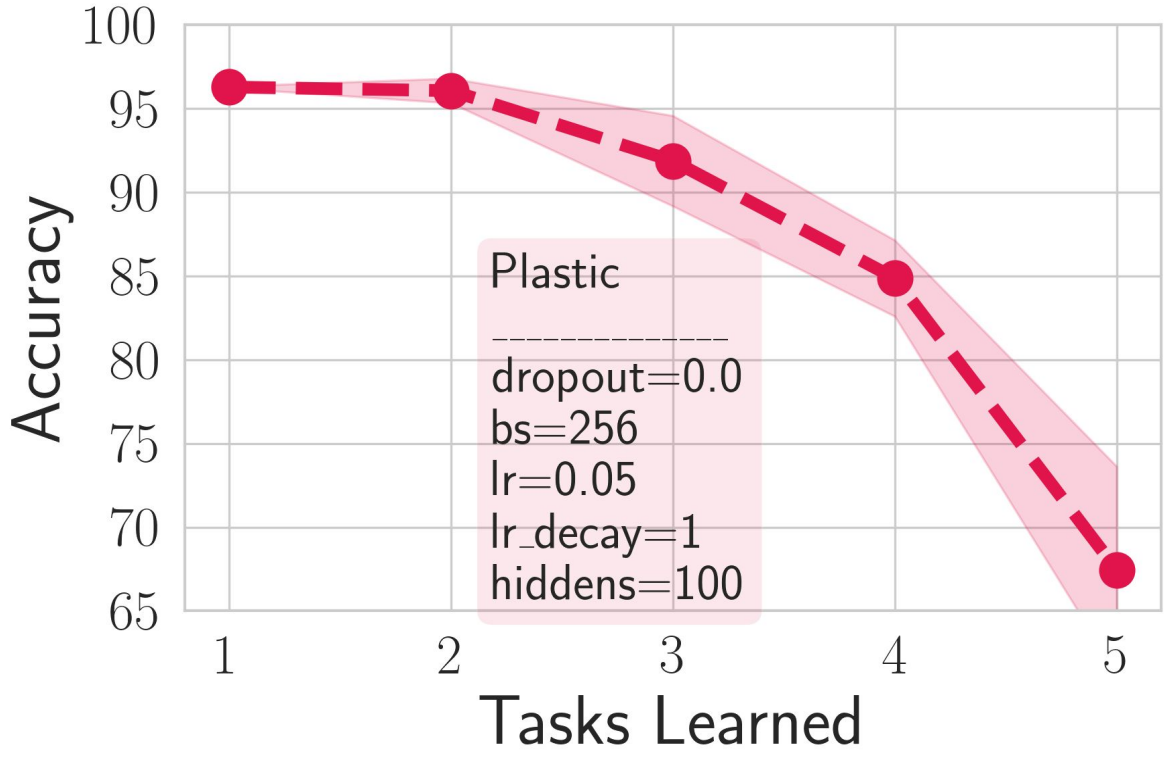
Q2: How to study the role of training regime?

Q3: How to define catastrophic forgetting? How to measure it?

Q4: What factors impact the catastrophic forgetting?

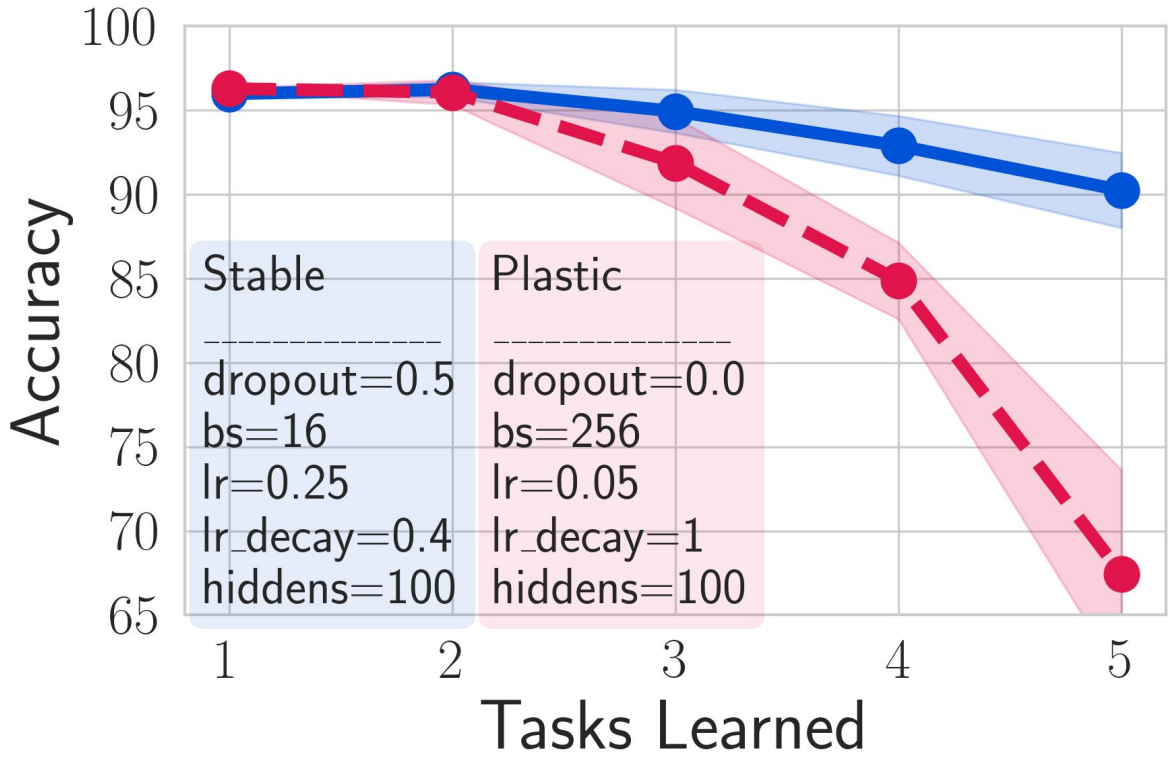
Q5: What is the impact of different training regimes on these factors?

Q1: How important is the training regime?



Q1: How important is the training regime?

- Same architecture
- Same tasks
- Different training regime



Q2: How do we study this phenomenon?

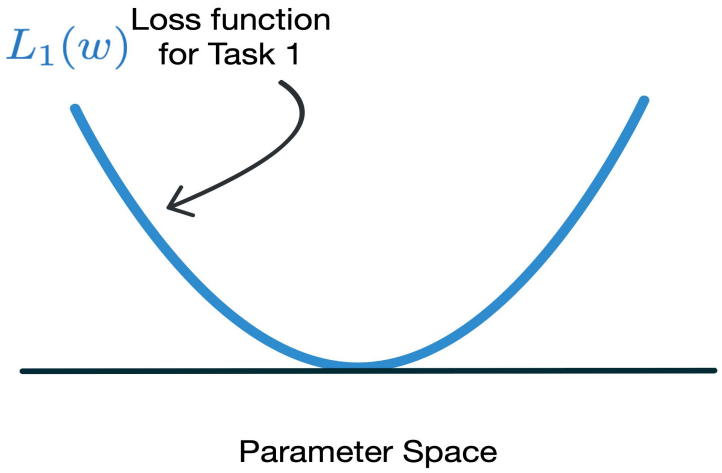
- To study the effect of training regime on catastrophic forgetting, we study the loss landscape of neural network during the continual learning experience

- To this end, we need a to define “catastrophic forgetting”.

Q3: How to measure forgetting? [Problem Setup]

w Network parameters

$L(w)$ Loss function of Task 1 w.r.t to parameters W

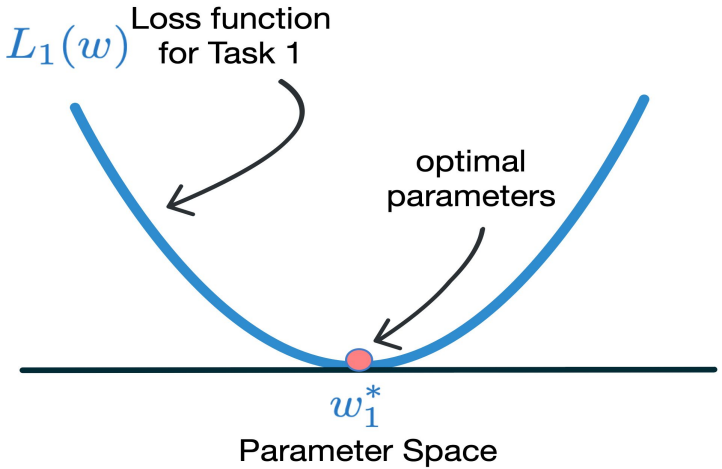


How to measure forgetting? [Problem Setup]

w Network parameters

$L(w)$ Loss function of Task 1 w.r.t to parameters W

w_t^* Minima found for task t after learning task t

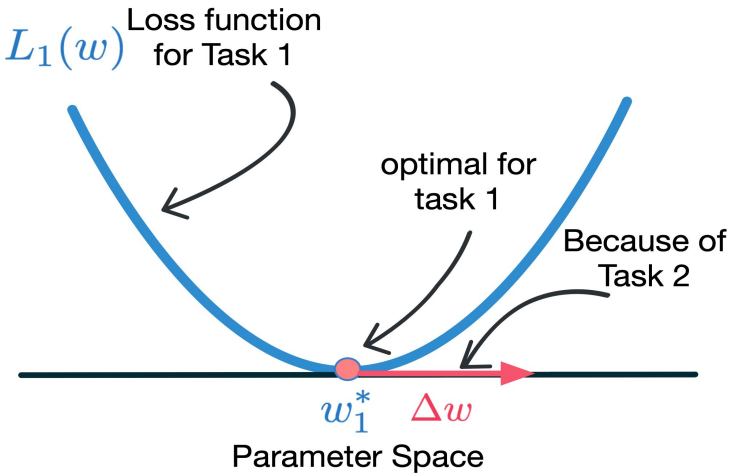


How to measure forgetting? [Problem Setup]

w Network parameters

$L(w)$ Loss function of Task 1 w.r.t to parameters W

w_t^* Minima found for task t after learning task t



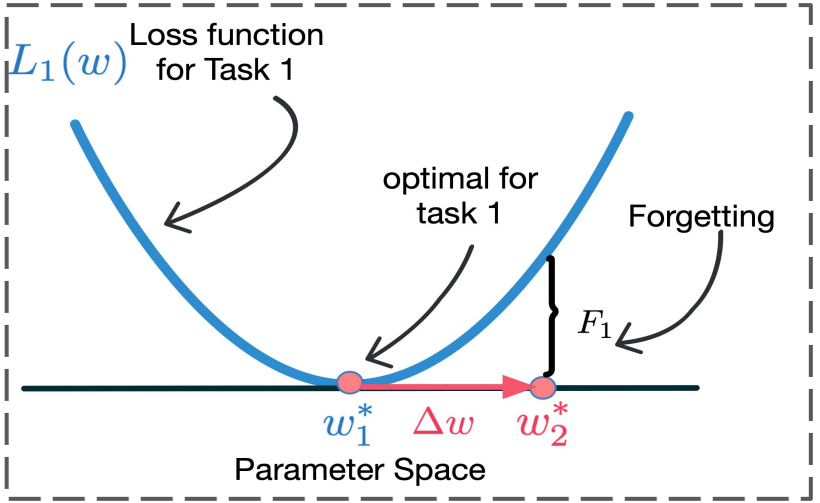
How to measure forgetting? [Problem Setup]

w Network parameters

$L(w)$ Loss function of Task 1 w.r.t to parameters W

w_t^* Minima found for task t after learning task t

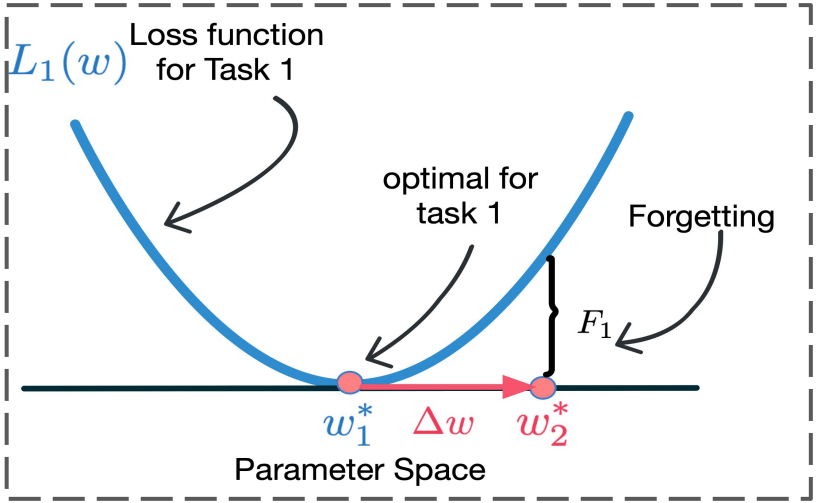
F_t Change in loss function of task t, after learning the next task



Definition of forgetting

In other words, forgetting is defined as:

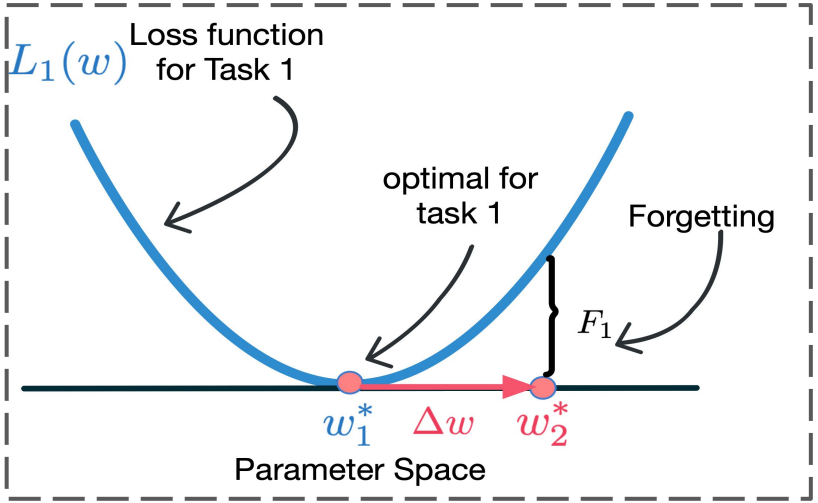
$$F_1 \triangleq L_1(w_2^*) - L_1(w_1^*)$$



Definition of forgetting

In other words, forgetting is defined as:

$$F_1 \triangleq L_1(w_2^*) - L_1(w_1^*)$$



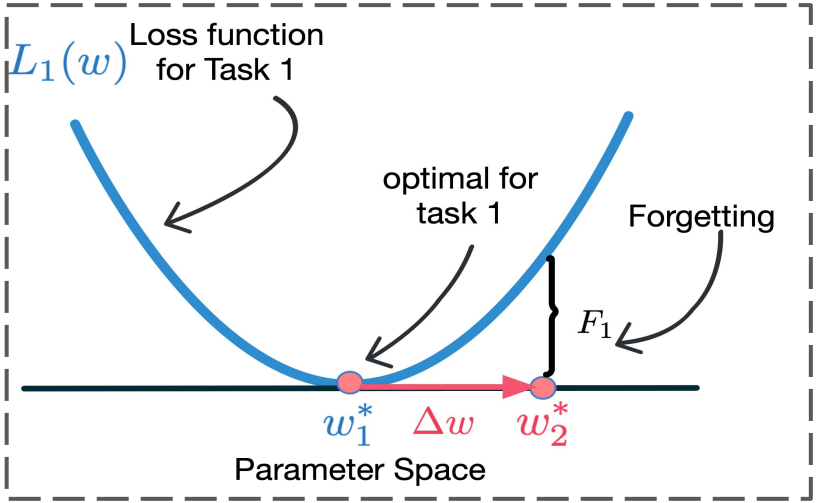
By Taylor's approximation:

$$L_1(w_2^*) - L_1(w_1^*) = (\Delta w)^\top \nabla L_1(w_1^*) + \frac{1}{2}(\Delta w)^\top \nabla^2 L_1(w_1^*)(\Delta w)$$

Definition of forgetting

In other words, forgetting is defined as:

$$F_1 \triangleq L_1(w_2^*) - L_1(w_1^*)$$



By Taylor's approximation:

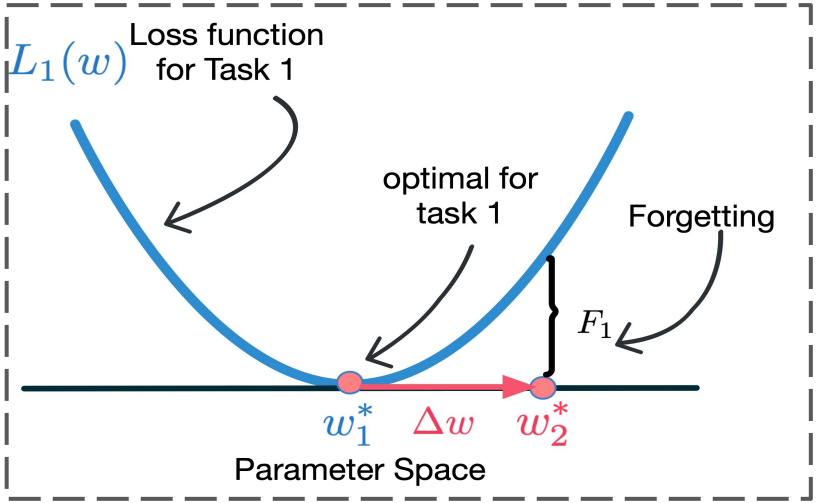
$$L_1(w_2^*) - L_1(w_1^*) = (\Delta w)^\top \nabla L_1(w_1^*) + \frac{1}{2}(\Delta w)^\top \nabla^2 L_1(w_1^*)(\Delta w)$$

Negligible gradient magnitude

Definition of forgetting

In other words, forgetting is defined as:

$$F_1 \triangleq L_1(w_2^*) - L_1(w_1^*)$$



By Taylor's approximation:

$$L_1(w_2^*) - L_1(w_1^*) = (\Delta w)^\top \nabla L_1(w_1^*) + \frac{1}{2}(\Delta w)^\top \nabla^2 L_1(w_1^*)(\Delta w)$$

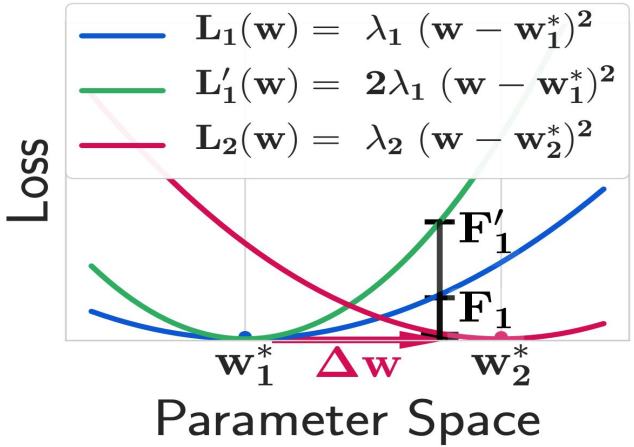
$$\leq \frac{1}{2}\lambda_1^{\max}\|\Delta w\|^2$$

Q4: What factors impact the amount of forgetting?

[Hypothesis]: *The amount of forgetting that a neural network exhibits from learning the tasks sequentially, correlates with the geometrical properties of the convergent points for each task.*
In particular, the wider these minima are, the less forgetting happens.

Q4: What factors impact the amount of forgetting?

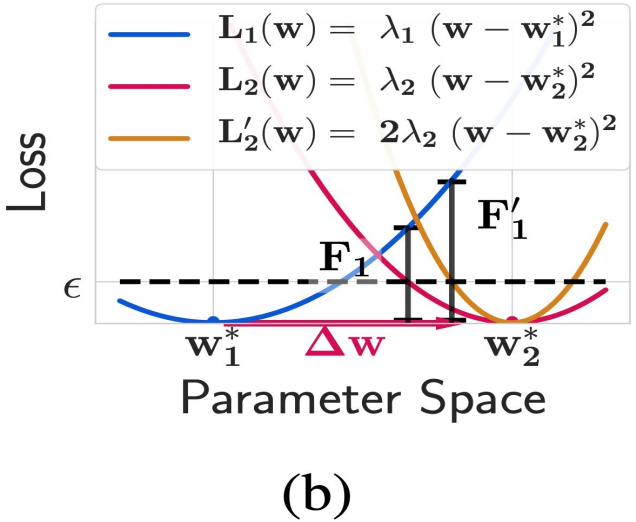
[Hypothesis]: *The amount of forgetting that a neural network exhibits from learning the tasks sequentially, correlates with the geometrical properties of the convergent points for each task.*
In particular, the wider these minima are, the less forgetting happens.



(a)

Q4: What factors impact the amount of forgetting?

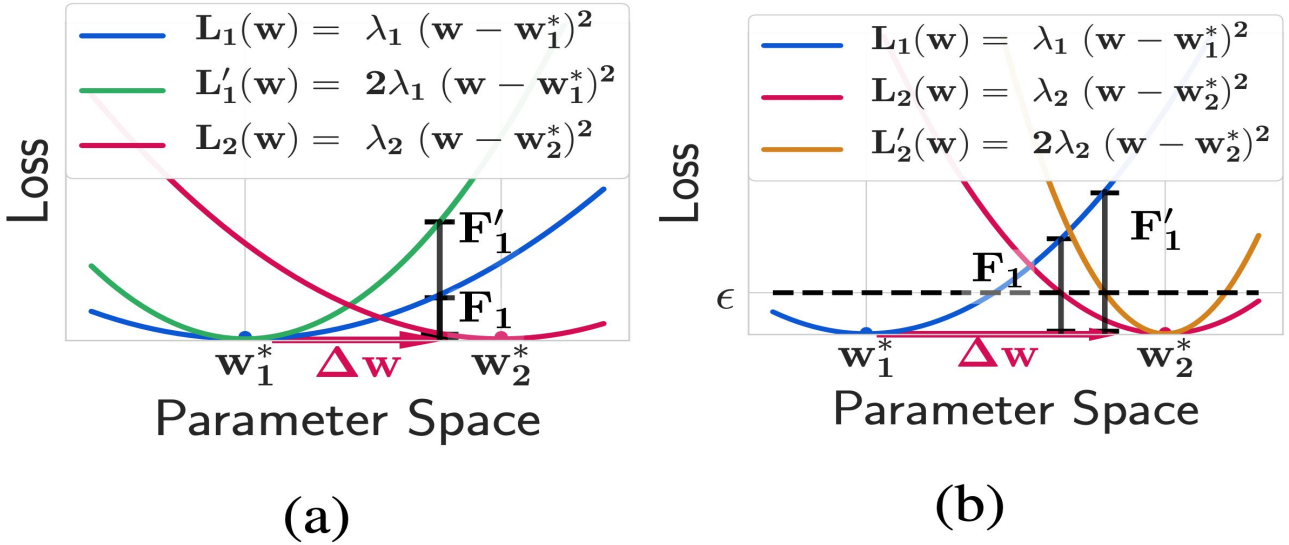
[Hypothesis]: *The amount of forgetting that a neural network exhibits from learning the tasks sequentially, correlates with the geometrical properties of the convergent points for each task.*
In particular, the wider these minima are, the less forgetting happens.



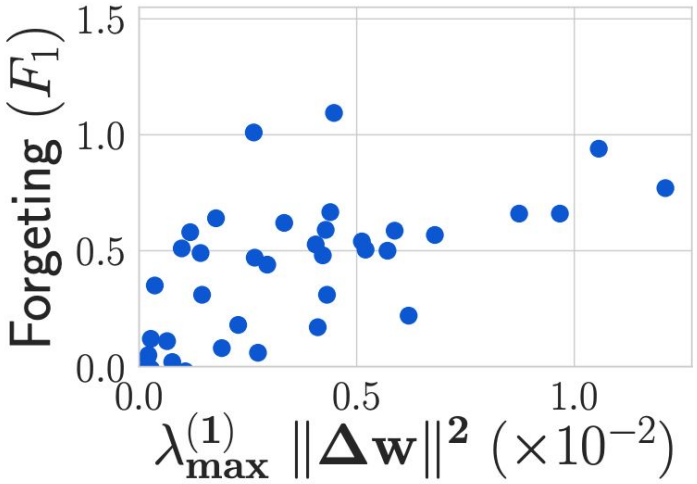
(b)

Q4: What factors impact the amount of forgetting?

[Hypothesis]: *The amount of forgetting that a neural network exhibits from learning the tasks sequentially, correlates with the geometrical properties of the convergent points for each task.*
In particular, the wider these minima are, the less forgetting happens.

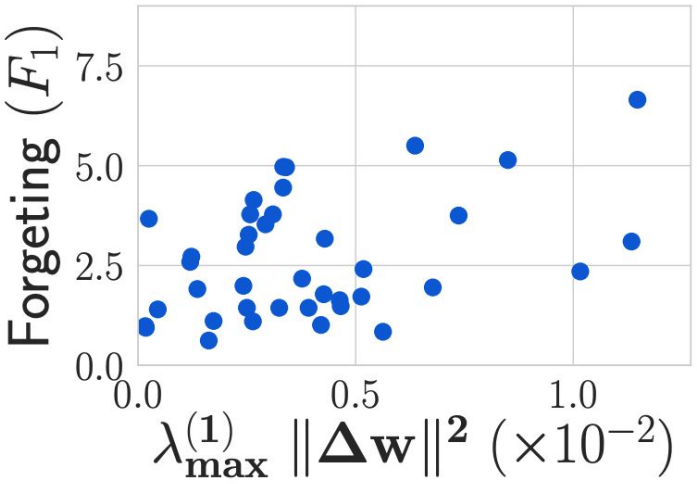


Empirical Verification



(c)

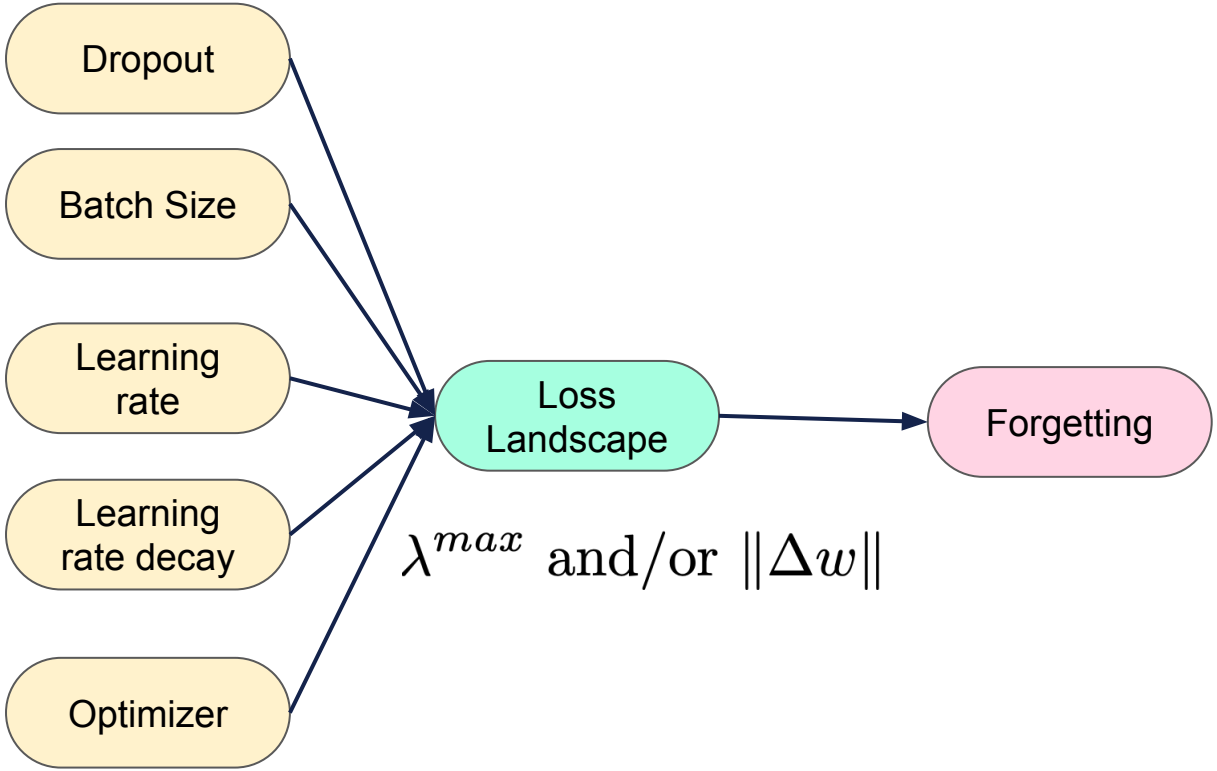
Rotated MNIST



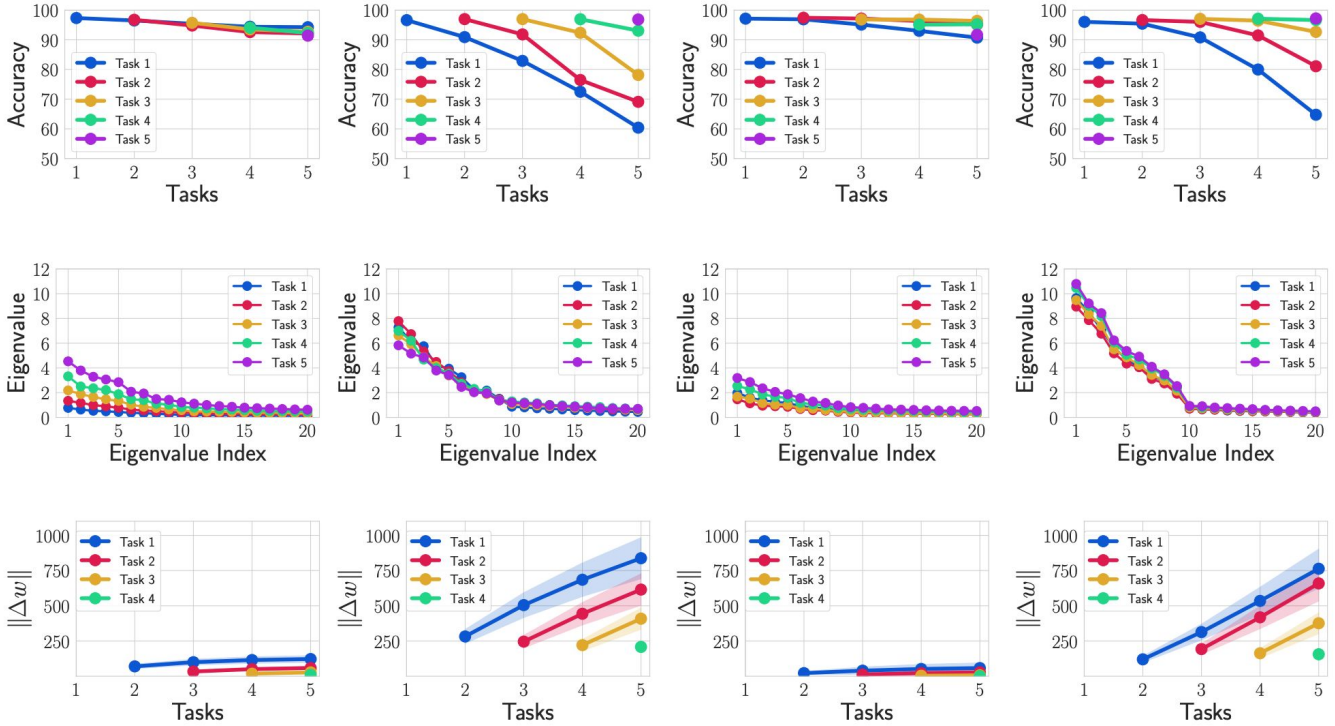
(d)

Permuted MNIST

Q5: Training Regimes: techniques affecting stability and forgetting



Experiment 1: Does these technique are really effective?



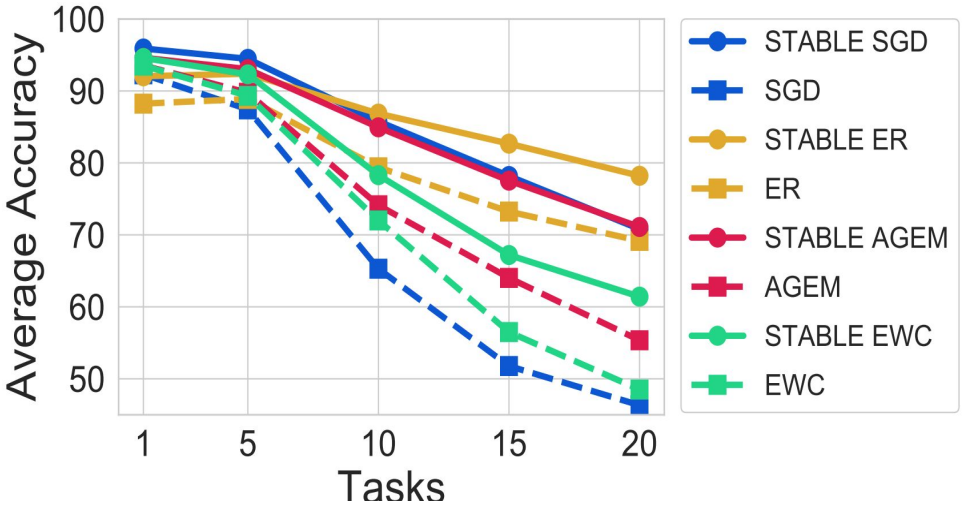
(a) Permuted - Stable (b) Permuted - Plastic (c) Rotated - Stable (d) Rotated - Plastic

Experiment 2: Comparison with other Methods

Method	Memoryless	Permuted MNIST		Rotated MNIST		Split CIFAR100	
		Accuracy	Forgetting	Accuracy	Forgetting	Accuracy	Forgetting
Naive SGD	✓	44.4 (±2.46)	0.53 (±0.03)	46.3 (±1.37)	0.52 (±0.01)	40.4 (±2.83)	0.31 (±0.02)
EWC	✓	70.7 (±1.74)	0.23 (±0.01)	48.5 (±1.24)	0.48 (±0.01)	42.7 (±1.89)	0.28 (±0.03)
A-GEM	✗	65.7 (±0.51)	0.29 (±0.01)	55.3 (±1.47)	0.42 (±0.01)	50.7 (±2.32)	0.19 (±0.04)
ER-Reservoir	✗	72.4 (±0.42)	0.16 (±0.01)	69.2 (±1.10)	0.21 (±0.01)	46.9 (±0.76)	0.21 (±0.03)
Stable SGD	✓	**80.1 (±0.51)**	**0.09 (±0.01)**	**70.8 (±0.78)**	**0.10 (±0.02)**	**59.9 (±1.81)**	**0.08 (±0.01)**
Multi-Task Learning	N/A	86.5 (±0.21)	0.0	87.3(±0.47)	0.0	64.8(±0.72)	0.0

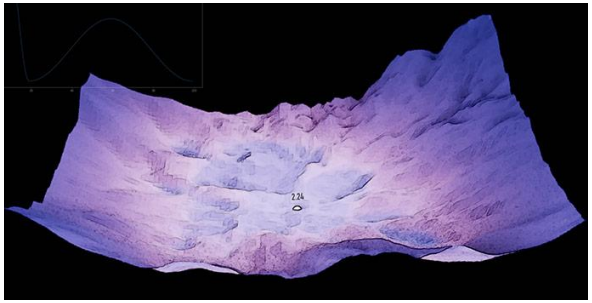
Experiment 3: Combining these techniques with SOTA methods

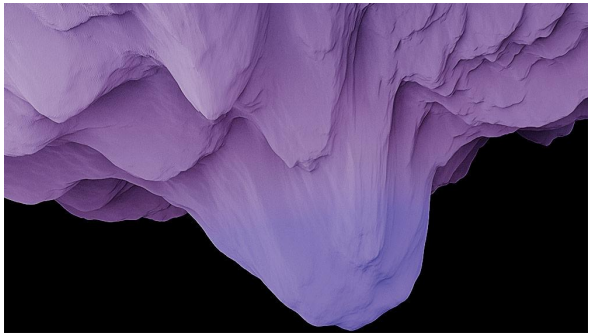
Method	Average Accuracy	Forgetting
SGD	46.3 (± 1.37)	0.52 (± 0.01)
Stable SGD	70.8 (± 0.78)	0.10 (± 0.02)
EWC	48.5 (± 1.24)	0.48 (± 0.01)
Stable EWC	61.4 (± 1.15)	0.30 (± 0.01)
AGEM	55.3 (± 1.47)	0.42 (± 0.01)
Stable AGEM	71.1 (± 1.06)	0.13 (± 0.01)
ER-Reservoir	69.2 (± 1.10)	0.21 (± 0.01)
Stable ER-Reservoir	78.2 (± 0.74)	0.09 (± 0.01)



Conclusion and future works

- Loss landscape
- Overparameterization, modern neural networks and loss surfaces
- Batch norm and other stabilization techniques
- Meta learning
- Bias of network architecture





SOLA: Continual Learning with Second-Order Loss Approximation

A loss function perspective

Yin, Farajtabar, Li. https://arxiv.org/abs/2006.10974, Continual Learning workshop at ICML 2020

Introduction

- Approximate the loss function by estimating its second-order Taylor expansion.
- Use them as surrogates added to the loss function of the current task.
- We only store information based on the entire training dataset, thus better protects privacy.

Good old Taylor expansion

Approximating the loss function around the minima of the first task

$$\widetilde{L}_1(w) = \widehat{L}_1(\widehat{w}_1) + (w - \widehat{w}_1)^\top \nabla \widehat{L}_1(\widehat{w}_1) + \frac{1}{2}(w - \widehat{w}_1)^\top \nabla^2 \widehat{L}_1(\widehat{w}_1)(w - \widehat{w}_1)$$

And minimize the approximated loss of first task plus the empirical loss of the second one

$$\tfrac{1}{2}(\widetilde{L}_1(w) + \widehat{L}_2(w))$$

SOLA (Second Order Loss Approximation) does not need to store any data!

Theoretical analysis

- A sufficient and worst-case necessary condition under which by conducting gradient descent on the approximate loss function, we can still minimize the actual loss function.
- Convergence analysis of the algorithm for both non-convex and convex loss functions.
- Our results imply that early stopping can be helpful in continual learning.

Experiments

Permuted MNIST, Rotated MNIST, Split MNIST

Dataset	P-MNIST	P-MNIST	R-MNIST	R-MNIST	R-MNIST	S-MNIST	S-MNIST
Model type	MLP	MLP	MLP	MLP	CNN	MLP	MLP
Model size	[10, 10]	[100, 100]	[10, 10]	[100, 100]	4-conv	[10, 10]	[100, 100]
Multi-task	91.8 ± 0.4	97.0 ± 0.1	91.4 ± 0.4	97.5 ± 0.1	98.8 ± 0.1	98.9 ± 0.3	99.3 ± 0.1
A-GEM	84.1 ± 1.1	93.2 ± 0.4	83.6 ± 1.0	92.6 ± 0.4	95.3 ± 0.3	91.2 ± 4.9	97.8 ± 0.4
Vanilla	69.2 ± 3.1	81.1 ± 1.6	76.8 ± 0.9	86.0 ± 0.5	89.5 ± 0.6	86.4 ± 6.6	97.2 ± 0.9
EWC	69.1 ± 3.7	80.2 ± 1.4	76.9 ± 1.0	86.1 ± 0.6	89.4 ± 0.7	87.7 ± 9.2	97.7 ± 0.8
OGD	68.9 ± 3.3	81.5 ± 1.7	81.1 ± 1.3	88.0 ± 0.7	89.5 ± 0.7	$\mathbf{97.1 \pm 1.8}$	98.8 ± 0.1
SOLA-exact	$\mathbf{90.0 \pm 0.9}$	–	$\mathbf{88.6 \pm 0.9}$	–	–	96.3 ± 3.0	–
SOLA-prox	86.2 ± 1.5	$\mathbf{87.8 \pm 0.6}$	86.5 ± 0.9	$\mathbf{90.4 \pm 0.5}$	$\mathbf{92.2 \pm 1.5}$	96.1 ± 2.5	$\mathbf{99.0 \pm 0.2}$

Split CIFAR

Model	Multi-task	A-GEM	Vanilla	EWC	OGD	SOLA-exact	SOLA-prox
CNN-2	75.9 ± 0.9	65.8 ± 2.1	57.2 ± 4.2	55.6 ± 4.6	56.5 ± 4.2	$\mathbf{62.0 \pm 5.4}$	59.4 ± 3.8
CNN-6	78.6 ± 1.4	68.1 ± 2.3	57.5 ± 4.6	57.7 ± 3.8	58.3 ± 4.8	–	$\mathbf{58.6 \pm 5.2}$
MLP[200, 200]	69.2 ± 0.5	66.1 ± 0.7	63.5 ± 1.6	63.8 ± 2.1	$\mathbf{65.8 \pm 1.2}$	–	55.7 ± 3.2

Conclusion

Future works of Continual Learning (CL)

A lot of open problems!

- Benchmarking and real datasets
- Generative and hybrid models in CL
- Multi-objective optimization and CL
- Sparsity and smart gating
- Attention and recurrent models
- Representation learning for orthogonalization
- Uncertainty and out of distribution
- Theoretical foundation and analysis of CL
- ...

Related to many fields!

- Learning over evolving distributions
- Dataset shift
- Domain adaptation
- Meta learning
- Lifelong learning
- Incremental learning
- Online and stream learning
- ...

Questions and Comments

Mehrdad Farajtabar

Feel free to contact: farajtabar@google.com for further questions!