CounterAdversarial Recall of Synthetic Observations

A *neuro-inspired* approach to foil gradient-based adversarial attacks

Emanuele Ballarin[†]

Supervised by: Prof. Luca Bortolussi[†]

CoSupervised by: Dr. Alessio Ansuini [‡]

Graduation session: M.Sc. in *Data Science and Scientific Computing*

30th October 2022

[†]AICPS ∈ Dept. of Mathematics ⊆ Univ. of Trieste | [‡]AREA Science Park

An *elevator pitch*

CARSO (*CounterAdversarial Recall of Synthetic Observations*) is a novel *deep learning* architecture and training/inference methodology for the improvement of *adversarial robustness* in *deep artificial neural networks*.

- Loosely inspired by high-level *neurocognitive* mechanisms;
- Targeted against *gradient-based*, *white-box* attacks;
- Significant, promising results so far; comprehensive testing still in early stages.

Deep Learning

Deep Learning is (probabilistic) *machine learning* with *deep artificial neural networks*, *operating in (strongly) overparametrised regime.*

OK, but what does it mean?

Machine Learning

> *A computer program is said to learn from experience E w.r.t. to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.*
>
> *(T. Mitchell, attributed – 1997)*

Allowing effective decoupling of:

- *model* (in our case: *artificial neural networks*)
- *learning algorithm* (in our case: *loss* function minimisation; approximate, iterative, gradient based[1])

[1] Not strictly necessary, though

Focus scenario for today: *supervised learning*.

Input space: $\mathbb{I} \sim \mathbb{R}^d$ – *What* needs to be classified, or their embedding;
Output space: $\mathbb{O}$ – Set of classes to partition $\mathbb{I}$ in (or *e.g. logits*, *etc.*).

Training set: $\{\mathbb{I} \times \mathbb{O}\} \supseteq \mathbb{T} = \{(\boldsymbol{x}_1, \xi_1), (\boldsymbol{x}_2, \xi_2), \ \ldots, \ (\boldsymbol{x}_n, \xi_n)\}$ – Known examples;
A classifier $\mathcal{N}_{\boldsymbol{\theta},\boldsymbol{h}} : \mathbb{I} \to \mathbb{O}$ dependent from parameters[1] ($\boldsymbol{\theta}$) and hyperparameters ($\boldsymbol{h}$);
A *loss function*: $\mathcal{L} = \mathcal{L}(\mathcal{N}_{\boldsymbol{\theta},\boldsymbol{h}}, \mathbb{T})$ encoding the *goodness* of the model on $\mathbb{T}$.

Goal of *training*

Find optimal $\boldsymbol{\theta}$, *i.e.* $\boldsymbol{\theta}^\star := \arg\min_{\boldsymbol{\theta}}(\mathcal{L}(\mathbb{T}|\boldsymbol{\theta}, \mathcal{N}_{\boldsymbol{\theta},\boldsymbol{h}}))$

[1] And statistics of the data; *computed* rather than *learned* optimisation-wise

Preliminarily, we consider the *vector-scalar* map $\mathcal{N}_{1n}\colon \mathbb{R}^d \to \mathbb{R}$ (*McCulloch-Pitts neuron*): an affine transformation followed by a nonlinear application, *i.e.*:

$$y = \mathcal{N}_{1n}(\boldsymbol{x}) = \mathcal{A}(b + \boldsymbol{w} \cdot \boldsymbol{x})$$

with learnable parameters $\boldsymbol{\theta} = (\boldsymbol{w}, b)$.

For a *vector-vector* map $\mathcal{N}_{1n}\colon \mathbb{R}^d \to \mathbb{R}^m$, we operate element-wise *w.r.t.* the output, defining a *neuron layer*, *i.e.*:

$$\boldsymbol{y} = L(\boldsymbol{x}) = \mathcal{A}(\boldsymbol{b} + \boldsymbol{W}\boldsymbol{x}) \ .$$

Such transformations can be composed, for (arbitrary[1]) increased expressive power, in a *fully connected*, *feedforward* artificial neural network:

$$\boldsymbol{y} = \mathcal{N}(\boldsymbol{x}) = L_n(L_{n-1}(\dots L_2(L_1(\boldsymbol{x})))) \ .$$

[1] Usually in the *function space density* sense – more generally *PAC-ly* – given *width*/*depth* and growing the other.

We are left with an optimisation problem in the (potentially *very high-dimensional*) space of $\boldsymbol{\theta}$, that is generally intractable. We resort to an *iterative*, *first-order*, approximate optimisation scheme. *E.g.* (*gradient descent with momentum*):

$$\boldsymbol{\theta'} \leftarrow \boldsymbol{\theta} - \lambda \boldsymbol{g_i} + \mu \boldsymbol{m}$$
$$\boldsymbol{m'} \leftarrow \boldsymbol{\theta'} - \boldsymbol{\theta} \,.$$

Usually, *minibatch-aggregation*[1] and further regularisation (*e.g.* penalisation at *loss* or *weight* level[2]) is employed.

[1] Often, *gradient averaging* over disjoint subsets of training data
[2] Source of the *proper* L_2 *vs. proper weight decay* divergence.

That was just a *sneak peek*, and there is much much more...

- Different architectural *inductive biases* (*e.g. convolutional-, graph- NNs*);
- More advanced optimisers (*e.g. EMA-based – à la Adam, nested,* $> 1^{st}$*-order*);
- Other regularisation strategies, *convergence/generalisation* aids (*i.e. BatchNorm*, *dropout*);
- Learning rate scheduling and hyperparameter tuning;
- Efficient gradient computation (*e.g.* via *BackProp*);
- *AutoDiff* internals, *etc.* ...

We showed just a *bare minimum* to allow what follows (*but if time allows...*).

Fast-forwarding to-date, *deep learning* is a remarkably powerful and mature paradigm, able to reach (super)human-level performance in selected *regression*, *classification*, data *generation* and *control* tasks.

88.9% bookcase

(P. Perdikaris, 2018)

🐿️ *Deep Learning* in a nutshell

🤔 The problem of *adversarial robustness*

🧠 *(relatively)* Robust neural systems: *brains*

🤖 CARS0: an *introspective artificial neural machine*

🧪 Experimental evaluation

🗣️ Discussion

🚀 Conclusion and future outlook

The last shown picture is an example of

Adversarial Input

An *input* is said to be *adversarial* to a machine learning system if it alters its reasonably expected behaviour[1]. Also called *adversarial attack*, stressing the intentional[2] crafting of it.

In the specific case of a classifier: produce a *misclassification*.

[1] Usually from the *P.o.V.* of the user(s).
[2] Which is not a strict requirement, though!

We live in times where a growing portion of even *high-stakes* decisions is delegated to autonomous systems (*e.g. HR* selection, insurance, health, fraud detection, *etc.* ...).
→ The example of *Lemonade* 🍋

Purely *technical* reasons

- Harden *ML*/*DL* systems against misuse and *input-tampering*;
- Assess (and *patch!*) behaviour where it the most fragile.

Legal / *ethical* / *social* reasons

- To ensure compliance with regulatory frameworks or coordinated initiatives thereof;
- Increase understanding, transparency, and societal trust.

Broader-reaching goals

- Use *robustness* as a lens through which to study *neurocognitive phenomena*.

An intuitive geometrical characterisation of *adversarial attacks* can be done in the light of the

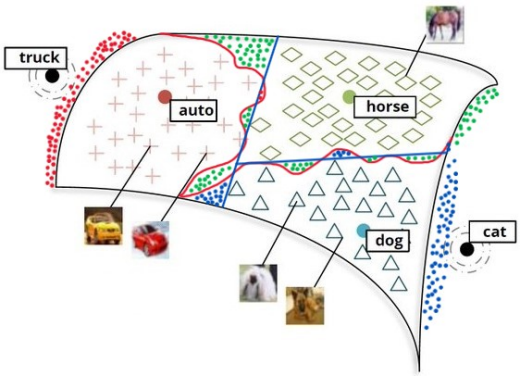
Manifold Hypothesis

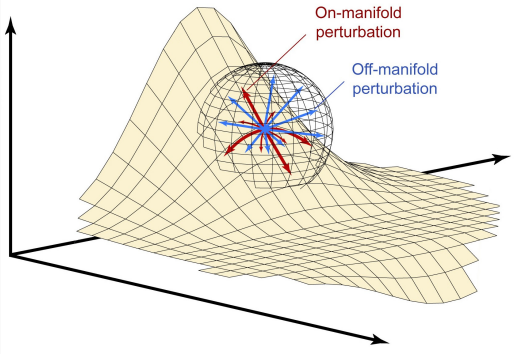
Natural high-dimensionally-coded data lie on (a) low-dimensional manifold(s), immersed within the high-dimensional allowed *code space*.

This partitions the *code space* in *four* regions:

1. Expected behaviour region (*on-manifold*);
2. Cross-boundary adversarial region (*on-manifold*);
3. *Natural* non-applicability region (*along manifold*);
4. 'Negative space' adversarial region (*off-manifold*).

'A picture is worth thousand words'. (*Here's* 2×10^3*!*)





We can always reformulate the problem of *adversarial inputs* as one of *adversarial perturbations*, *i.e.*

$$\boldsymbol{x}_{\text{adversarial}} := \boldsymbol{x}_{\text{legitimate}} + \boldsymbol{p}$$

leading to the following

Definition: ϵ-*perturbative adversarial attack against classifier* $\mathcal{N}$ *in* $x_0 \in \mathbb{I}$, *w.r.t.* $||\cdot||$

Any $\boldsymbol{x^\star} := \boldsymbol{x_0} + \boldsymbol{p} \mid \mathcal{N}(\boldsymbol{x^\star}) \neq \mathcal{N}(\boldsymbol{x_0})$ and $||\boldsymbol{p}|| < \epsilon$

Minimal systematics:

- *black box* vs. *white box*;
- *targeted* vs. *untargeted*.

Needless to say: *the world is not so simple; however...*

White-box scenario

- Direct, gradient-based (*e.g.* `FGSM` & derived);
- Iterative, gradient-based (*e.g.* `PGD`, `DeepFool`);
- Specific response-elicitation methods (*e.g. k-pixel* attack, *noisy* `FGSM`).

Black-box scenario

- *Gradient-free* optimisation schemes;
- Direct-sampling *generative* methods (*e.g.* `AdvGAN(++)`);
- *White-box* methods towards surrogate models (*e.g. Tramèr ensemble*).

Definition: *Universal Attack* (scheme)

A scheme for generating adversarial attacks able to reach any point within the (chosen norm-induced) ball of radius ϵ ($\forall\epsilon$) centred in any *legitimate* input point. Related: *information optimal attack*.

That of *adversarial robustness* research is a typical open, *ladder-and-fence* scenario…
With an even greater variability in defences:

- *Adversarial training*: augment $\mathbb{T}$ with correctly-labelled adversarial inputs (*e.g.* `IAT`);
- Adversarial detection: identify and handle anomalous inputs (*e.g.* `GAN`-*discriminator anomaly detection*);
- Adversarial purification: recover *clean* inputs from perturbed ones (*e.g.* `PuVAE`, `DefenseGAN`, *diffusion-based purification*);
- Inference-time defences: *e.g. filter* based, *test-time augmentation* based, *etc.* ;
- Robustly-structured learning (*e.g. Parseval Networks*);
- Paradigmatic shifts (*e.g. Bayesian* NNs).

No optimal, universal defence! Many *case-by-case* results, many *trade-offs*, practically no *robust-by-design* applicable solution.

A remark

But... have *you* ever experienced an *adversarial(-like) phenomenon?*

🐿️ *Deep Learning* in a nutshell

🤔 The problem of *adversarial robustness*

🧠 *(relatively)* Robust neural systems: *brains*

🤖 CARS0: an *introspective artificial neural machine*

🧪 Experimental evaluation

🗣️ Discussion

🚀 Conclusion and future outlook

Focus on vision

It is safe to say that examples similar in form to *adversarial attacks* for *NNs* are yet to be discovered for *e.g.* human subjects.

Some related phenomena, however, do *probably* exist (within *vision*):

- Retinal response-elicitation (*e.g. impossible colours*; incomplete evidence);
- Attention retargeting via saliency shaping;
- Amygdala-mediated (*semantic*) attention retargeting;
- Fast, *adversarial* elicitation of the early visual pathway (anecdotal).

...and up to *optical illusions*, with a stretch.

Yet, *brains* may be the *only* practical realisation of a system with the *robustness* properties we look for…

💡 A guiding idea

Is it possible to loosely inform the development of *robust* DL systems with (grossly simplified, idealised) descriptions of neurocognitive phenomena?

Getting inspiration from the ideas of *recall of acquired information*, and *introspection* as *thought about thought*:

- Role of *recall* as a *comparison tool* for newly acquired information;
- Role of *recall* (and the aware *anticipation* of it) in learning and *gap-filling* memories
  → *Forward testing* phenomenon
  → *Repeated testing* phenomenon
- *Liminality* and hippocampal dynamics.

[...] I gazed—and gazed—but little thought
What wealth the show to me had brought:

For oft, when on my couch I lie
In vacant or in pensive mood,
They flash upon that inward eye [...]

(W. Wordsworth – I Wandered Lonely as a Cloud)

⚠ Beware!

The *modelling* that follows has no claim of *biological plausibility* whatsoever, at this stage! This would be *added value*, though – and in interesting research direction!

🐿️ *Deep Learning* in a nutshell

🤔 The problem of *adversarial robustness*

🧠 *(relatively)* Robust neural systems: *brains*

🤖 CARS0: an *introspective artificial neural machine*

🧪 Experimental evaluation

🗣️ Discussion

🚀 Conclusion and future outlook

Our synthetic *thought process* (focus on *classification*):

1. All processing within a (non-stochastic) *NN* is deterministic at inference time;
2. Different outputs from *clean* and successfully *perturbed* inputs must *leave a different trace* within the representation of a classifier;
3. We can use such representation for *adversarial detection*
   $\rightarrow$ Unsatisfactory!
4. *And what about purification*? We can inform a cAE with such representation;

Remark!

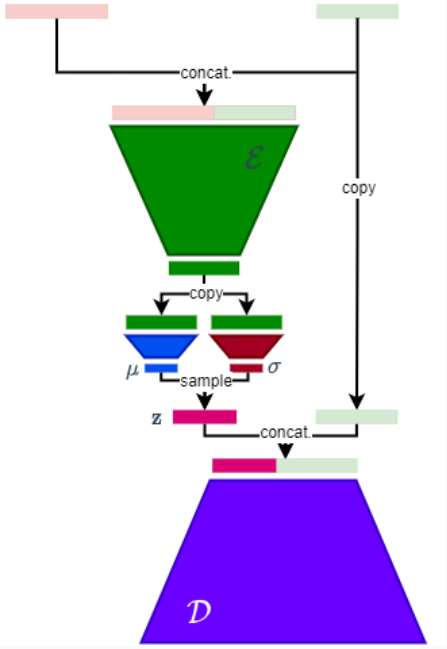
If the purified input is evaluated by the *very same* classifier, the *representation-as-hidden-layers* and the *representation-as-*cAE*-input* induce competing gradients for a *gradient-based* adversary!

Our synthetic *thought process* (cont.):

5. At inference time, the representation is more than enough to reconstruct the input;
6. We can learn a **cVAE** and sample from the specific purified-input space!
7. Samples can be later classified and the outputs aggregated.

Remark!

As a *side effect*, this adds another layer of defence: a *sampling-invariance* for the adversary to overcome!



Variational Autoencoders are *artificial neural architectures* able to sample from probability densities in the form

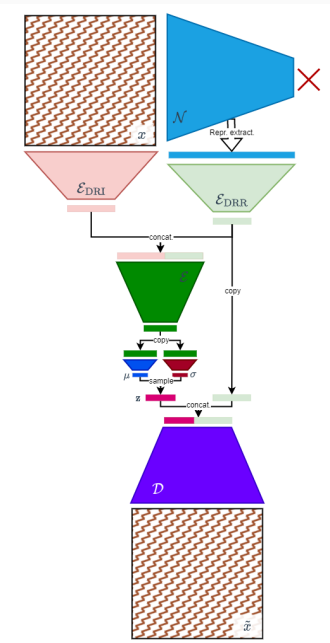
$$\boldsymbol{x} \sim \boldsymbol{p}(\tilde{\boldsymbol{x}}|\phi_1, \ldots, \phi_{s\in\mathbb{N}}) \,.$$

They operate in *standard* autoencoding settings, with, additionally:

- Code-space sampling;
- The approximate reparametrisation: $\boldsymbol{x} \approx \tilde{\boldsymbol{x}} = \mathcal{D}_{\boldsymbol{\theta_D}}(\boldsymbol{c} \sim \boldsymbol{p}_{\text{latent}}(\boldsymbol{c}|\mathcal{E}_{\boldsymbol{\theta_E}}(\boldsymbol{x})))$
- A *loss* constraining the code distribution to a given structure: $\mathcal{L}_{\text{VAE}} \coloneqq \mathcal{L}_{\text{AE}} + \text{KL}(\boldsymbol{p}_{\text{latent}}, \boldsymbol{p}_{\text{lobs}})$

Conditional VAEs extend the sampling to conditional distributions of the kind:

$$\tilde{\boldsymbol{x}}_{\text{r.v.}} \sim \boldsymbol{p}(\tilde{\boldsymbol{x}}_{\text{r.v.}}|\boldsymbol{x}_{\text{c.v.}}, \phi_1, \ldots, \phi_{s\in\mathbb{N}})$$



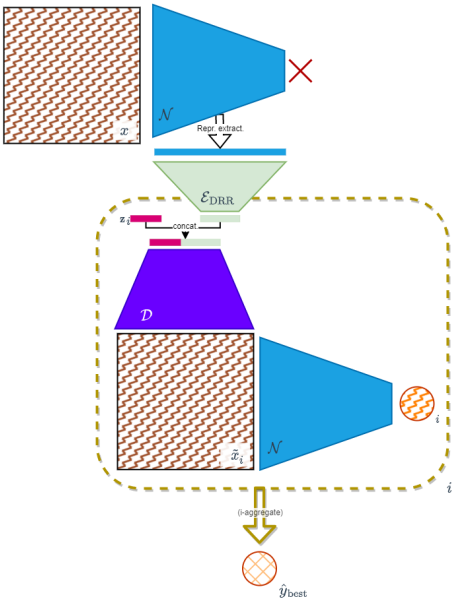
TL;DR: Just a *fancy* **cVAE**!

Given an *adversarially-pretrained* classifier (for the problem of interest, and according to a given *threat model*):

- First *classification pass* for representation extraction;
- Pre-encoding and rebalancing of input & representation;
- As in **cVAE**, aiming at *purified input* reconstruction from any input.

Requirements

A dataset of *clean/attacked* inputs to the classifier is needed (but no labels!). Threat models may differ.



TL;DR: Condition, sample, classify, aggregate!

Given the same *adversarially-pretrained* classifier, the just-trained *representation pre-compressor* and *decoder*:

- First *classification pass* for representation extraction;
- Representation *pre-encoding*;
- Repeated sampling of *candidate purified inputs*;
- Second classification pass on such reconstructions, for actual classification;
- Aggregation of results.

The difference between a great idea and an idea that works is the part in which you make it work.

(M. Budinich – 'Introduction to the theory of NNs' course; final remarks before the exam)

🐿️ *Deep Learning* in a nutshell

🤔 The problem of *adversarial robustness*

🧠 *(relatively)* Robust neural systems: *brains*

🤖 CARS0: an *introspective artificial neural machine*

🧪 Experimental evaluation

🗣️ Discussion

🚀 Conclusion and future outlook

General goal: accurate – but *narrow-scoped* – investigation; beyond *proof-of-concept*, not extensive. Focus on image classification tasks.

Foreseen attacks: whole-dataset **FGSM** ($||\cdot||_2$), **PGD** ($||\cdot||_\infty$); various strengths ($\epsilon = 0.15$, $\epsilon = 0.30$ [1]).

Unforeseen attacks: whole-test-set **DeepFool**, various strengths ($\epsilon = 0.15$, $\epsilon = 0.30$, $\epsilon = 0.50$); **FGSM** ($||\cdot||_2$), **PGD** ($||\cdot||_\infty$), fixed strength ($\epsilon = 0.50$).

Classifier: *Fully-connected feedforward* with 3 hidden layers, **Mish** activation, *BatchNorm*, ~ 0.15-dropout. Representation size: 290; trainable parameters: ~ 17.5k.
Trained with **RAdam**, $\lambda_{\text{init}} = 0.05$; 300 epochs, reducing *l.r.* on plateaus.
Loss: *one-hot class similarity* L_2.

[1] On normalised data, $\epsilon = 0.30$ is considered a sort of *reasonable* upper limit!

cVAE parts: *Fully-connected feedforward*, *deep* but as shallow as possible within reasonable reconstruction similarity. Trainable parameters: $\mathcal{E}_{\text{DRI}}$: ~ 0.5m, $\mathcal{E}_{\text{DRR}}$: ~ 61k, $\mathcal{E}$: ~ 80k, μ/σ layers: ~ 5.3k each, $\mathcal{D}$: ~ 2.1m. *Leaky* ReLU act. Trained with RAdam, $\lambda_{\text{init}} = 0.001$; 300 epochs, reducing *l.r.* on plateaus. Loss: *pixelwise binary cross-entropy*. Inference-time number of samples: 1500, *mode-class* aggregated.

Other hyperparameters: Batch size: 256; Number of neurons in hidden layers: *funnel-like, programmatically generated*. Very little experimentation *w.r.t.* hyperparameter tuning.

‘There’s nothing like the sheer power of numbers to scrub away layers of confusion and contradiction.’ (— S. Levitt , economist)

Attack / Defence (*adv. acc.%*)	None	IAT	CARSO
None	**98.40**	97.17	96.72
FGSM $\|\|\cdot\|\|_2$, $\epsilon = 0.15$	12.09	91.89	**93.62**
FGSM $\|\|\cdot\|\|_2$, $\epsilon = 0.30$	01.21	76.94	**86.43**
(U) FGSM $\|\|\cdot\|\|_2$, $\epsilon = 0.50$	01.00	12.29	**13.59**
PGD $\|\|\cdot\|\|_\infty$, $\epsilon = 0.15$	01.60	90.54	**93.44**
PGD $\|\|\cdot\|\|_\infty$, $\epsilon = 0.30$	06.85	71.26	**86.27**
(U) PGD $\|\|\cdot\|\|_\infty$, $\epsilon = 0.50$	*20.66*	*11.67*	**38.38**
(U) DF $\|\|\cdot\|\|_\infty$, $\epsilon = 0.15$	00.66	90.25	**95.06**
(U) DF $\|\|\cdot\|\|_\infty$, $\epsilon = 0.30$	00.00	60.54	**93.31**
(U) DF $\|\|\cdot\|\|_\infty$, $\epsilon = 0.50$	00.00	00.78	**71.34**

Many ablation studies have been performed, both during model development and after the final architecture was completely determined. Namely:

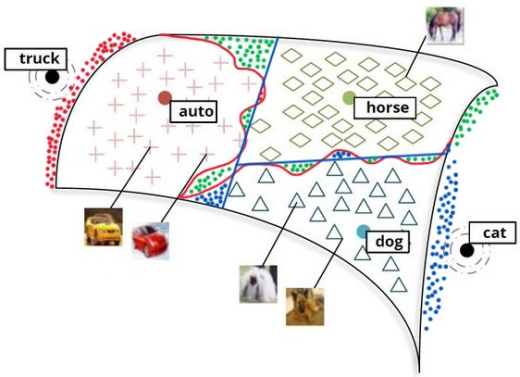
- *On the necessity of adversarial training* $\rightarrow \sim 15\text{-}25\%$ *adversarial accuracy* $(\sim 100\times)$;
- On the number of purified samples $\rightarrow > 1500$;
- On the number of layers in the **cVAE** network $\rightarrow$ *by incremental construction*;
- On the number of training epochs $\rightarrow$ Indeed, might be as low as 60 (FCN) and 100 (**CARSO**), but requiring accurate scheduling;
- On the choice of the optimiser $\rightarrow$ No improvement, slight loss increase at convergence.

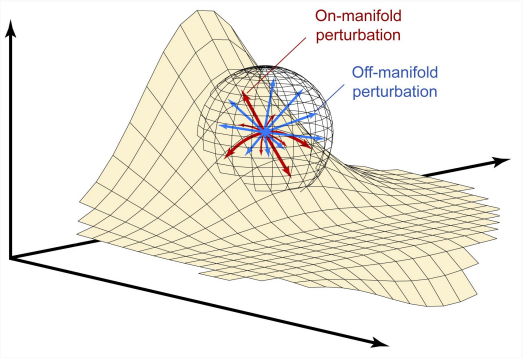
- 🐿️ *Deep Learning* in a nutshell
- 🤔 The problem of *adversarial robustness*
- 🧠 *(relatively)* Robust neural systems: *brains*
- 🤖 CARS0: an *introspective artificial neural machine*
- 🧪 Experimental evaluation
- 🗣️ Discussion
- 🚀 Conclusion and future outlook

Within the scope of the experimental analysis performed so far, we consider the results obtained to be *moderately-to-very* positive.

- A *clean accuracy toll* is imposed by the method *w.r.t. IAT*. Yet, this is to be generally expected, and slight in magnitude;
- Against *foreseen attacks*: significant – but not large – increase in *adversarial accuracy*;
- Against *unforeseen attacks*: very solid performance, clearly beyond *foreseen attacks/defences* transferability. *Innate robustness*

Speculatively: the result of a combined, synergistic effect. However, the lens of the *data manifold hypothesis* may give a more precise analysis...





CARSO acts mainly as an *on manifold re-projector*!

Time: a hidden cost?

Regardless of model performance, the *training time* required for the CARSO portion of the alone is $\sim$ equivalent to that of `IAT` for the classifier. This results in a twofold increase in training time.

Inference time sees a ~ 1500-fold increase.

Note, however, that...

- Overall training time is still (*much*, even *very much*; see `DefenseGAN` *e.g.*) shorter than most *better-than-*`IAT` approaches!
- *W.r.t.* inference time, the method was originally targeted at *high-stakes*, scenarios where such trade-off is acceptable. In case of realtime scenarios, also thanks to increased robustness, one can resort to *stream thinning* (or *fast-vs-slow* systems) if CARSO is deemed important.

🐿️ *Deep Learning* in a nutshell

🤔 The problem of *adversarial robustness*

🧠 *(relatively)* Robust neural systems: *brains*

🤖 CARS0: an *introspective artificial neural machine*

🧪 Experimental evaluation

🗣️ Discussion

🚀 Conclusion and future outlook

We talked about CARSO – a novel framework devised to foil *gradient-based adversarial attacks*, specifically targeted at image classification – showing noteworthy improvements upon IAT, a strong contribution to *off-manifold-to-on-manifold reprojection*, and solid *innate robustness*.

Experimental scope can be broadened, though:

- Proper, meticulous hyperparameter tuning;
- Different architectures;
- More complex datasets;
- Different *media* (text, sequences, *etc.* ...);
- Broader attack *coverage* (e.g. against *adaptive attacks*);
- Comparison with additional defences.

The work required to develop and assess CARSO evoked suggestions reaching far longer and broader than expected. Chiefly, in order of increasing conceptual distance…

- The idea that *adaptive* defences may exist, explicitly steering their behaviour on the basis of the geometric properties of inputs or attacks faced. CARSO might even one (though embryonic, immature) of this kind.
- Weight-agnostic layers operating at the *feature-specific* (or, traditionally, *architecture/dataset-specific*) level – able to produce *zero-gradient* in expectation, without masking it.
- The possibility of informing the development (or… the *training*?!) of *deep learning* architectures with (the information contained in) neural activity recordings – for the sake of one or the other, or both. And, why not: *live-subject* recordings! 🐁

Toward Next-Generation Artificial Intelligence: Catalyzing the NeuroAI Revolution

By Anthony Zador, Blake Richards, Bence Ölveczky, Sean Escola, Yoshua Bengio, Kwabena Boahen, Matthew Botvinick, Dmitri Chklovskii, Anne Churchland, Claudia Clopath, James DiCarlo, Surya Ganguli, Jeff Hawkins, Konrad Koerding, Alexei Koulakov, Yann LeCun, Timothy Lillicrap, Adam Marblestone, Bruno Olshausen, Alexandre Pouget, Cristina Savin, Terrence Sejnowski, Eero Simoncelli, Sara Solla, David Sussillo, Andreas S. Tolias, Doris Tsao

Abstract: *Neuroscience has long been an important driver of progress in artificial intelligence (AI). We propose that to accelerate progress in AI, we must invest in fundamental research in NeuroAI.*

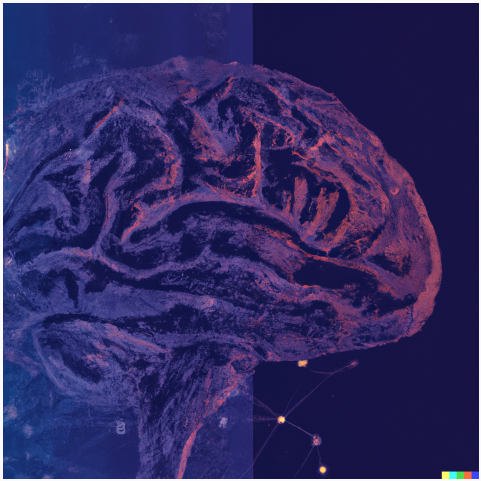
(Zador et al., 2022 – *ArXiv, abs/2210.08340*)

Eugenio Culurciello

Oct 19 · 11 min read

How can we build an artificial brain from our knowledge of the human brain?

(Eugenio Culurciello, 2022 – *Medium.com*, see: https://cutt.ly/culurciello_brain_2022)



[https://ballarin.cc/carsocode]