Causality in Neural Networks

Inverse mechanisms

Maciej Żelaszczyk

March 11, 2020

PhD Student in Computer Science
Division of Artificial Intelligence and Computational Methods
Faculty of Mathematics and Information Science

m.zelaszczyk@mini.pw.edu.pl

Warsaw University
of Technology

Independent mechanisms

Mechanisms:

- Humans are able to adapt to new domains with little to no retraining.
- This might be because we rely on mechanisms that are independent of the particular domain.
- For instance, people are able to recognize distorted images from the get-go.
- It can be hypothesized that these mechanisms are modular, reusable and broadly applicable.

Independent mechanisms

The *independent mechanisms* (IM) assumption:

- The causal generative process of a system's variables is composed of autonomous modules that do not inform or influence each other.

Independent mechanisms

Let us consider variables $x_1, \ldots, x_d$. If their joint density is Markovian w.r.t. a directed acyclic graph $\mathcal{G}$, we can write:

$$p(\mathbf{x}) = p\left(x_1, \ldots, x_d\right) = \prod_{j=1}^{d} p\left(x_j | \mathrm{pa}_{\mathcal{G}}^{j}\right) \tag{1}$$

where $\mathrm{pa}_{\mathcal{G}}^{j}$ denotes the parents of variable x_j in the graph.

- In the general case, for a given joint density function, we can find many graphs (decompositions) of such form.
- If the edges of $\mathcal{G}$ denote direct causation, then $\mathcal{G}$ is called a *causal* graph and each conditional probability $p\left(x_j | \mathrm{pa}_{\mathcal{G}}^{j}\right)$ can be understood as a *causal mechanism* generating x_j from its parents.
- The presented factorization is a *generative* model in the sense of describing an actual physical *generative* process.

Independent mechanisms

Consequences of the IM assumption:

- The causal conditionals are autonomous modules that do not influence or inform each other.
- Knowledge of one mechanism does not contain information about another one.
- Changes in one mechanism do not affect the other mechanisms - invariance.
- An intervention in one mechanism does not impact other ones.
- If we change $p\left(x_j|\mathrm{pa}_{\mathcal{G}}^j\right)$, other mechanisms $p\left(x_i|\mathrm{pa}_{\mathcal{G}}^i\right), i \neq j$ do not change.
- Consider that this is not true for other factorizations that do not capture the causal structure.

Independent mechanisms

Machine learning models expressed in terms of causal mechanisms could:

- Facilitate transfer learning, domain adaptation, generalization.
- Provide modularity and the opportunity to train parallel components, which could be recombined into larger systems.
- Offer more interpretability.
- Increase sample efficiency.
- Help in overcoming catastrophic forgetting.

Elephant in the room

Given a causal graph learning can be extremely efficient, but:

- Nobody gives us this graph.
- Exhaustive search is not feasible.
- Methods like the *maximum width spanning tree* algorithm can be used together with measures based on mutual information.
- None of them seem to work for really large problems.
- We would be interested to learn the causal mechanisms from data without blowing up.

Making it more concrete

We could focus on a particular class of causal mechanisms and the ability to learn them from data:

- Let us consider image transformations.
- We would like to identify inverse transformations from data.
- We do not know the transformations in advance.
- We do not know which transformation produces which image.
- We do not have a pairing between the base image and the transformed image.
- We do not even see the base images corresponding to the seen transformed images.
- We only get a sample from the reference distribution and a sample of other transformed images.

Formalization

- Consider a canonical distribution $P(\mathbf{X})$ of image data, where $\mathbf{X} \in \mathbb{R}^d$.
- Define N measurable functions $M_1, \ldots, M_N : \mathbb{R}^d \to \mathbb{R}^d$. These functions (transformations) represent independent causal mechanisms.
- Based on the transformations, we can define the distributions $Q_1, \ldots, Q_N$, where $Q_j = M_j(P)$.
- At training time, we receive a dataset $\mathcal{D}_Q = (x_i)_{i=1}^n$ drawn from a mixture of $Q_1, \ldots, Q_N$ and a dataset $\mathcal{D}_P$ sampled from the canonical distribution.
- We want to identify $M_1, \ldots, M_N$ and learn the inverse mappings $M_1^{-1}, \ldots, M_N^{-1}$.

Approach

Let us approach the problem of learning the inverse mappings by applying a training procedure with:

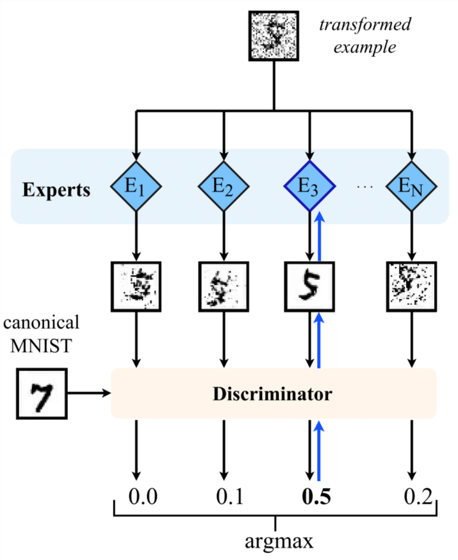
- $N^{'}$ functions $E_1, \ldots, E_{N'}$ parametrized by $\theta_1, \ldots, \theta_{N'}$ - these functions will be called *experts*.
- In general $N \neq N^{'}$.
- Maximize the objective function $c : \mathbb{R}^d \rightarrow \mathbb{R}$ with the property that c takes high values on the support of the canonical distribution P , and low values outside.
- Each $x^{'} \in \mathcal{D}_Q$ is fed to all the experts.
- The values $c_j = c(E_j(x^{'}))$ are computed for all experts and the winning expert E_{j^*} is selected based on $j^* = \text{argmax}_j(c_j)$.
- The parameters θ_{j^*} of the winning expert are updated to maximize $c(E_{j^*}(x^{'}))$. We train c as well.

Approach

The objective function for the experts can be formulated as:

$$\theta_1^*, \ldots, \theta_{N'}^* = \underset{\theta_1^*, \ldots, \theta_{N'}^*}{\operatorname{argmax}} \mathbb{E}_{x' \sim Q} \left(\max_{j \in \{1, \ldots, N'\}} c(E_{\theta_j}(x')) \right) \tag{2}$$

Approach



Source: [Parascandolo et al., 2018]

Adversarial training

The general training procedure can be cast in an adversarial framework:

- Each expert is represented by a *generator* network G_j conditioned on the input image rather than a noise vector.
- The output of each generator is fed into a *discriminator* network D.
- For a given input x, the winning generator G_{j^*} is updated with backpropagation while other generators remain frozen.
- The discriminator D is trained against all the generators.

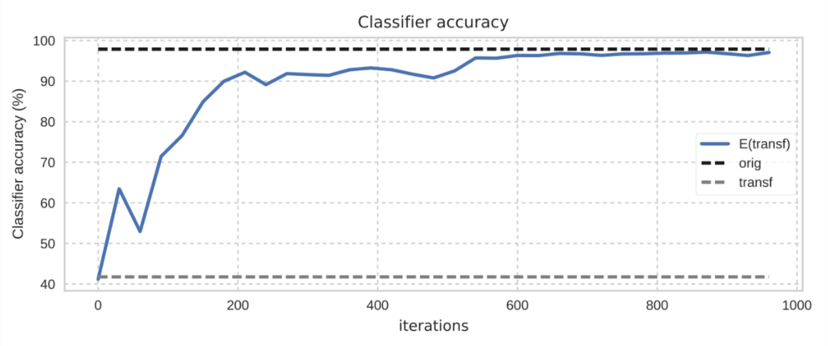
The discriminator is trained to maximize:

$$\max_{\theta_D} \left(\mathbb{E}_{x \sim P} \left[\log \left(D_{\theta_D}(x) \right) \right] + \frac{1}{N'} \sum_{j=1}^{N'} \mathbb{E}_{x' \sim Q} \left[\log \left(1 - D_{\theta_D}(E_{\theta_j}(x')) \right) \right] \right) \tag{3}$$

Neural network details

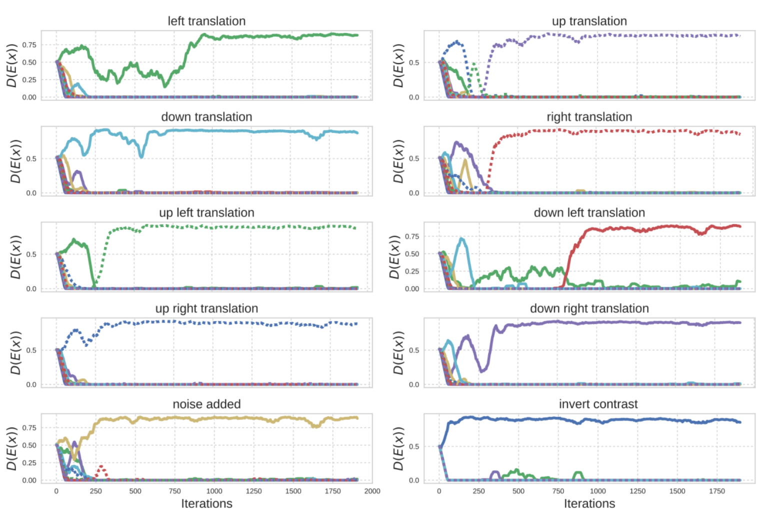
- Each expert: CNN with 5 convolutional layers, 32 filters per layer of size 3×3, ELU activations, batch normalization and zero padding.
- Discriminator: CNN with average pooling every 2 convolutional layers, a growing number of filters and a fully-connected layer of size 1024 as the last hidden layer.
- Trained with Adam with default hyperparameters.
- *Approximate identity initialization*: following a random initialization, the experts are trained on transformed data only to approximate identity transformations.

Results on MNIST



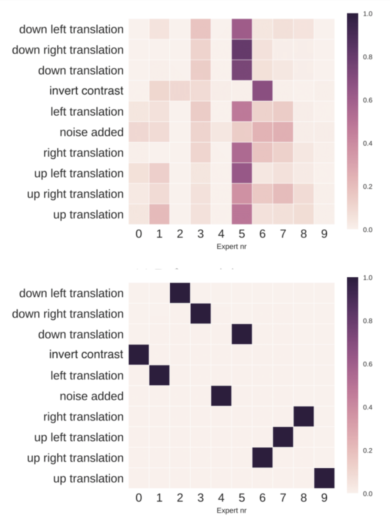
Source: [Parascandolo et al., 2018]

Results on MNIST



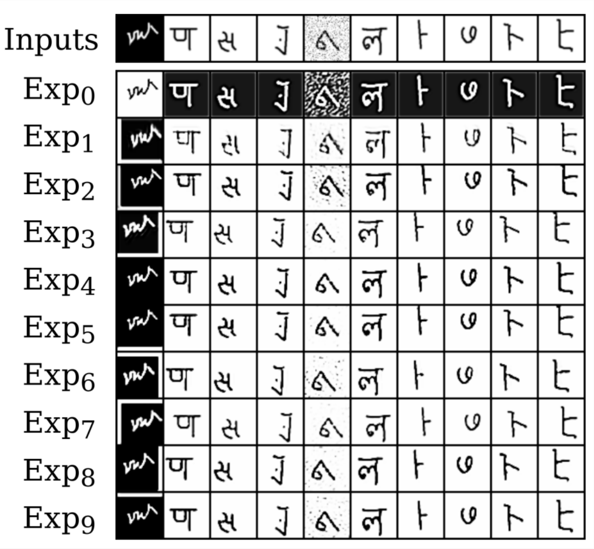
Source: [Parascandolo et al., 2018]

Results on MNIST



Source: [Parascandolo et al., 2018]

Results on MNIST



Source: [Parascandolo et al., 2018]

Parascandolo, G., Kilbertus, N., Rojas-Carulla, M., and Schölkopf, B. (2018).
Learning independent causal mechanisms.
In Dy, J. and Krause, A., editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4036–4044, Stockholmsmässan, Stockholm Sweden. PMLR.