Neural Networks for Learning Counterfactual G-Invariances from Single Environments

"Fixing the Image Rotation Problem"

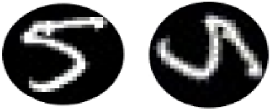
J. Setpal

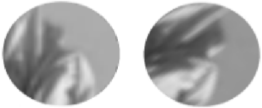
January 23, 2024

MACHINE LEARNING @ PURDUE

Neural Networks Aren't Rotationally Robust.

$\mathbf{Q}_1$: Do you think that a CNN trained on a distribution of the left image *should* classify the right image as the same class for each of these pairs?





Neural Networks Aren't Rotationally Robust.

$\mathbf{Q}_1$: Do you think that a CNN trained on a distribution of the left image *should* classify the right image as the same class for each of these pairs?

$\mathbf{A}_1$: Definitely!

Neural Networks Aren't Rotationally Robust.

$\mathbf{Q}_1$: Do you think that a CNN trained on a distribution of the left image *should* classify the right image as the same class for each of these pairs?

$\mathbf{A}_1$: Definitely!

$\mathbf{Q}_2$: In practice, does this actually happen?

Neural Networks Aren't Rotationally Robust.

$\mathbf{Q}_1$: Do you think that a CNN trained on a distribution of the left image *should* classify the right image as the same class for each of these pairs?

$\mathbf{A}_1$: Definitely!

$\mathbf{Q}_2$: In practice, does this actually happen?
$\mathbf{A}_2$: Nope – all these images were misclassified.

Neural Networks Aren't Rotationally Robust.

Q$_1$: Do you think that a CNN trained on a distribution of the left image *should* classify the right image as the same class for each of these pairs?

A$_1$: Definitely!

Q$_2$: In practice, does this actually happen?
A$_2$: Nope – all these images were misclassified.

Q$_3$: How can we fix this?

Neural Networks Aren't Rotationally Robust.

Q$_1$: Do you think that a CNN trained on a distribution of the left image *should* classify the right image as the same class for each of these pairs?

A$_1$: Definitely!

Q$_2$: In practice, does this actually happen?
A$_2$: Nope – all these images were misclassified.

Q$_3$: How can we fix this?
A$_3$: Data Augmentation (boring)

Neural Networks Aren't Rotationally Robust.

$\mathbf{Q}_1$: Do you think that a CNN trained on a distribution of the left image *should* classify the right image as the same class for each of these pairs?

$\mathbf{A}_1$: Definitely!

$\mathbf{Q}_2$: In practice, does this actually happen?
$\mathbf{A}_2$: Nope – all these images were misclassified.

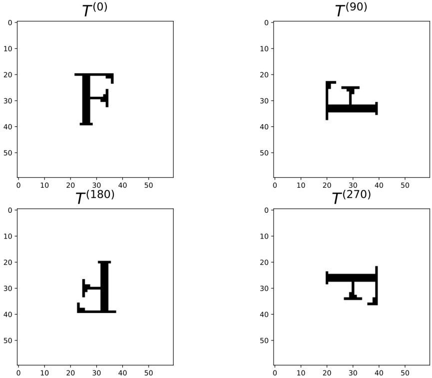
$\mathbf{Q}_3$: How can we fix this?
$\mathbf{A}_3$: Data Augmentation (boring), **G-Invariant Transformations** (fun)!

Images as Transformations

We can visualize the image rotations as affine matrix transformations:

$$G_{rot} \equiv \{ T^{0^\circ}, T^{90^\circ}, T^{180^\circ}, T^{270^\circ} \} \quad (1)$$

$$x_{new} = Tx_{orig}; T \in G_{rot} \quad (2)$$



Mathematical Formulation

Defining the transformations as a **group** gives us *guarantees* we can exploit to ensure **invariance** to those transformations.

Mathematical Formulation

Defining the transformations as a **group** gives us *guarantees* we can exploit to ensure **invariance** to those transformations.

Formally, we create an embedding layer to achieve the following:

$$\sigma(w^T x + b) \stackrel{def}{=} \sigma(w^T \boldsymbol{T} x + b); \boldsymbol{T} \in G_{rot} \tag{3}$$

Mathematical Formulation

Defining the transformations as a **group** gives us *guarantees* we can exploit to ensure **invariance** to those transformations.

Formally, we create an embedding layer to achieve the following:

$$\sigma(w^T x + b) \stackrel{def}{=} \sigma(w^T \boldsymbol{T} x + b); \boldsymbol{T} \in G_{rot} \tag{3}$$

This is only possible if we can find a transformation $\bar{T}$ such that:

$$\bar{T}(\boldsymbol{T} x) = \bar{T} x;\ \text{same as } \bar{T} x_{new} = \bar{T} x_{orig}; \tag{4}$$

Mathematical Formulation

Defining the transformations as a **group** gives us *guarantees* we can exploit to ensure **invariance** to those transformations.

Formally, we create an embedding layer to achieve the following:

$$\sigma(w^T x + b) \stackrel{def}{=} \sigma(w^T \boldsymbol{T} x + b); \boldsymbol{T} \in G_{rot} \tag{3}$$

This is only possible if we can find a transformation $\bar{T}$ such that:

$$\bar{T}(\boldsymbol{T} x) = \bar{T} x; \text{ same as } \bar{T} x_{new} = \bar{T} x_{orig}; \tag{4}$$

Lemma: We can find $\bar{T}$ using the *Reynold's Operator*.

$$\bar{T} = \frac{1}{|G|} \sum_{g \in G} g \tag{5}$$

Mathematical Formulation

Defining the transformations as a **group** gives us *guarantees* we can exploit to ensure **invariance** to those transformations.

Formally, we create an embedding layer to achieve the following:

$$\sigma(w^T x + b) \stackrel{def}{=} \sigma(w^T \boldsymbol{T} x + b);\ \boldsymbol{T} \in G_{rot} \tag{3}$$

This is only possible if we can find a transformation $\bar{T}$ such that:

$$\bar{T}(\boldsymbol{T}x) = \bar{T}x;\ \text{same as } \bar{T}x_{new} = \bar{T}x_{orig}; \tag{4}$$

Lemma: We can find $\bar{T}$ using the *Reynold's Operator*.

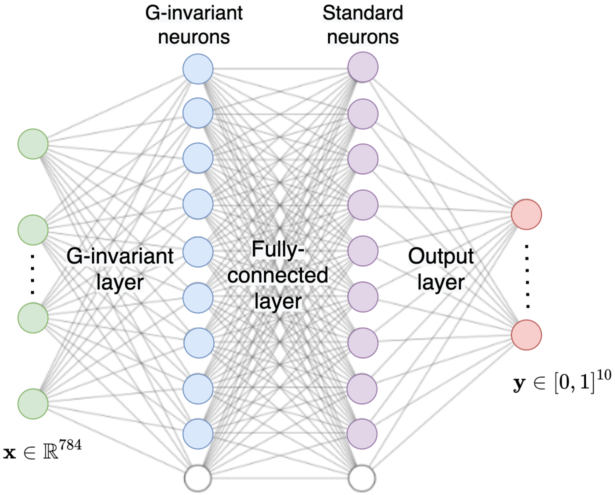
$$\bar{T} = \frac{1}{|G|} \sum_{g \in G} g \tag{5}$$

Finally, we construct our group invariant layer:

$$h_{inv} = \sigma(w^T \bar{T} x + b) \tag{6}$$

Let's Demonstrate!

Here's what the final architecture looks like:



Thank you!

Hopefully, this was cool!

Paper: `https://arxiv.org/abs/2104.10105/`

Slides: `https://cs.purdue.edu/homes/jsetpal/slides/gti.pdf`

Notebook: `https://cs.purdue.edu/homes/jsetpal/nb/gti.ipynb`

Presentation: `https://www.youtube.com/watch?v=znJsaCGiu10`