Machine Learning and Physics
Winter Semester 2024/2025

Exercise Sheet #11

Due: 27.01.24 14h00

Peter Lippmann

General Regulations.

- Please hand in your solutions in groups of two (preferably from the same tutorial group). Submissions by a single person alone will not be corrected.
- Your solutions to theoretical exercises can be either handwritten notes (scanned), or typeset using LaTeX. For scanned handwritten notes please make sure that they are legible and not too blurry.
- For the practical exercises, the data and a skeleton for your jupyter notebook are available at https://github.com/sciai-lab/mlph_w24. Always provide the (commented) code as well as the output, and don't forget to explain/interpret the latter. Please hand in your notebook (`.ipynb`), as well as an exported pdf-version of it.
- Submit all your files in the Übungsgruppenverwaltung, only once for your group of two. Specify all names of your group in the submission.

1 Decomposition of Cartesian tensors

(a) Check that the definition of a group representation $\rho : G \to GL(V)$ implies that $\rho(e) = I$, where e is the identity element of the group and I the identity matrix on the vector space V, on which the representation acts. Secondly, show that the definition implies that $\rho(a^{-1}) = \rho(a)^{-1}$. (2 pts)

The following exercises are specific to rotations in 3D Euclidean space, which from the group of SO(3). Cartesian tensors of order n transform under a rotation matrix $R \in \mathbb{R}^{3\times3}$ in the following way:

$$T'_{i_1\dots i_n} = R_{i_1 j_1} \dots R_{i_n j_n} T_{j_1\dots j_n} \quad \text{(using Einstein sums)}. \tag{1}$$

(b) A Cartesian tensor T_{ij} of order 2 transforms like $T' = RTR^{\mathrm{T}}$ (written in dimension notation this corresponds to the representation $\underline{3} \otimes \underline{3}$). Show that a Cartesian tensor T can be decomposed into a 1-dimensional, a 3-dimensional and a 5-dimensional subrepresentation (i.e. $\underline{5} \oplus \underline{3} \oplus \underline{1}$), according to

$$T = \underbrace{\frac{1}{3}\mathrm{Tr}(T) I_3}_{=:A} + \underbrace{\frac{1}{2}(T - T^{\mathrm{T}})}_{=:B} + \underbrace{\frac{1}{2}\big(T - T^{\mathrm{T}} - \frac{2}{3}\mathrm{Tr}(T) I_3\big)}_{=:C} \tag{2}$$

where I_3 is the 3×3 identity matrix. Concretely, you have to show that transforming A, B, C individually (e.g. $A \to RAR^{\mathrm{T}}$) yields the same A', B', C' as transforming $T \to T'$ and decomposing T' according to Eq. (2). *Hint: work in matrices rather than components.* (3 pts)

(c) In the lecture, it was shown that A transforms according to the trivial representation, since $\mathrm{Tr}(RTR^{\mathrm{T}}) = \mathrm{Tr}(T)$. In this exercise, you will show that the antisymmetric part B transforms like a vector. For that, argue that B can be parametrized as $B_{ij} = \epsilon_{ijk} v_k$ using the fully antisymmetric tensor ϵ_{ijk}. Then, show that the transformation of $B \to RBR^{\mathrm{T}}$ implies that v transforms like a vector. *Hint: Work in components and use the following determinant relation: $\epsilon_{i_1\dots i_n} M_{i_1 j_1} \dots M_{i_n j_n} = \det(M) \epsilon_{j_1\dots j_n}$ for a general $n \times n$ matrix M (Einstein sum implied).* (3 pts)

(d) In terms of dimensions, provide the decomposition of $\underline{3} \otimes \underline{3} \otimes \underline{3}$ into irreps using the general rule of

$$\underline{2l_1+1} \otimes \underline{2l_2+1} = \bigoplus_{L=|l_1-l_2|}^{l_1+l2} \underline{2L+1}. \tag{3}$$

(Apologies, in the lecture there was a typo on the RHS. Please correct it in your notes.) (2pts)

(e) Generalizing the decomposition from d), argue that the decomposition of a Cartesian tensor of rank n, denoted by $\underbrace{\underline{3} \otimes ... \otimes \underline{3}}_{n \text{ times}}$ contains exactly one irreducible representation with $l = n$. (1 pt)

(f) Bonus Argue that the irreducible subrepresentation with $l = n$ corresponds to the symmetric and traceless part of the decompositon. For that, count the number of independent components in a fully symmetric traceless tensor with n indices. A tensor is fully symmetric if $T_{...i...j...} = T_{...j...i...}$ for all pairs of indices and traceless if any contraction of two indices is zero, i.e. $\sum_i T_{...i...i...} = 0$ (for all pairs of indices). *Hint: You may find the solution to the so called stars and bars problem useful.* (3 pts)

2 Equivariant neural networks

Consider a geometric graph neural network (GNN) f, which takes as input a point cloud $P = \{x_i\}$ (a set of points located at $\mathbf{x}_i \in \mathbb{R}^3$) and produces a single output $f(P)$ for the whole point cloud.

The message passing layers ensure that the output is invariant w.r.t. permutations of the input points. Furthermore, assume that only relative positions $\mathbf{x}_i - \mathbf{x}_j$ between nodes are used in the message passing so that $f(P)$ is also invariant w.r.t. global translations of the point cloud.

(a) Show that for a general equivariant function $g(x)$ that the function output has the same symmetries as the function input. An input is said to exhibit a symmetry if it fulfills $\rho(g)x = x$ for some transformation g. (1 pt)

(b) Consider a geometric GNN as above that in addition is exactly SO(3)-equivariant. Argue that if you have a point cloud P which evenly samples an ellipsoid as input and want to predict an output that transforms like a vector that this vector can only be the zero vector. (2 pts)

SO(3)-Equivariant neural networks based on the irreducible representations can be build using the e3nn library (`https://e3nn.org/`). The e3nn library handels the transformations of spherical tensors via Wigner-D matrices and performs the decompositon of tensor products into irreps of SO(3) using the Clebsch-Gordan coefficients.

(c) Install e3nn and use the documentation (in particular the user guide on Irreps and convolution) as a resource to solve the following tasks (1 pt each, if not statetd otherwise):

1. Verify that the Wigner-D matrices for $l = 1$ is the rotation matrix itself, showing that $l = 1$ is indeed the vector representation.
2. Check at the examples of $l = 2, 3, 4$ that the dimensions of D-Wigner matrices are $(2l+1)\times(2l+1)$.
3. Verify for $l = 1, 2, 3, 4$ that spherical harmonics are equivariant functions (i.e. that transforming the input commutes with transforming the output), use that the outputs of $Y^l = (Y^l_{-l}, Y^l_{-l+1}, ..., Y^l_l)$ transform via the corresponding D-Wigner matrix $D^l(R)$.
4. Use the tensor product (`o3.FullyConnectedTensorProduct`) between two irreps of $l = 1$ and visualize it using `.visualize()` to verify that $\underline{3} \otimes \underline{3} = \underline{5} \oplus \underline{3} \oplus \underline{1}$.
5. Numerically compute the tensorproduct between to vectors $\mathbf{v} = (1, 2, 3)^{\mathrm{T}}, \mathbf{w} = (4, 5, 6)^{\mathrm{T}}$. Use a random rotation to verify that the tensor product is equivariant.

6. (2 pts) Consider the fully connected tensor product between two representations which are direct sums of irreps, namely `64x0e+32x1o+16x2e` and `32x1o+16x2e+8x3o` and visualize it. The fully connected tensorproduct computes the tensor product and its decompositon for all pairs of irreps. Use Eq. (3) to verify the set of output representations that e3nn has computed. Note: e and o define the transformation behavior under reflections (which we can ignore if we are focusing on pure rotations).

(7 pts)

3 Representations of electron densities for machine learning

Given a molecule with atoms positioned at $\mathbf{x}_i \in \mathbb{R}^3$, you would like to find a representation of the electron density $n(\mathbf{x})$ that can serve as a suitable input to a machine learning model that predicts molecular properties directly from the electron density. For that, given a finite set of atom-centered basis functions $w_\mu(\mathbf{x})$, one may approximate the density as $n(\mathbf{x}) = \sum_\mu p_\mu w_\mu(\mathbf{x})$. The idea of machine learning OFDFT is to use a neural network to predict the energy of an atom configuration as a function of the electron density. The predicted energy (or rather its gradient obtained via backprop) is then used optimize the electron density iteratively through gradient descent.

(a) Assuming that the functions $w_\mu(\mathbf{x})$ can be negative in some regions of space, the electron densities may become negative, i.e. unphysical, during density optimization. Do you have a proposal how to deal with this problem? (2 pts)

(b) If one were to use a different density representation such as $n(\mathbf{x}) = \left(\sum_\mu p_\mu w_\mu(\mathbf{x})\right)^2$ the density is guaranteed to be non-negative. Can you think of practical problems which may arise using this representation? (2 pts)