# Deep learning experiments with special Vogan diagrams

In this folder we make some experiments concerning special Vogan diagrams, where special is defined below. Let $\mathfrak{g}$ be a real semisimple Lie algebra of rank $\ell$ with Vogan diagram having indices of painted nodes $P$ and set of positive roots $\Delta^+$. Let $\{\gamma_1, \dots, \gamma_\ell\}$ be the set of simple roots and $\{\varphi_1, \dots, \varphi_\ell\}$ be the set of fundamental dominant weights. We assign to each root $\alpha$ a coefficient $\varepsilon_\alpha$ which is $-1$ if $\alpha$ is compact and 1 otherwise. Define the vector $\eta = \sum_{\alpha \in \Delta^+} \varepsilon_\alpha \alpha$ and consider the vector

$$\varphi_P := \eta - 2 \sum_{\alpha \in \mathrm{span}\{\gamma_i | i \in P^c\} \cap \Delta^+} \alpha,$$

which may be expressed in the basis of fundamental dominant weights as $\sum_{i \in P} a_i \varphi_i$, with $a_i \in \mathbb{Z}$. We say that a Vogan diagram is *special* if the $a_i$'s are all negative, zero or positive. In particular, the diagram is said to be *symplectic general type*, *symplectic Calabi-Yau* and *symplectic Fano* respectively. For more details about the theory beyond special Vogan diagrams see [1].

Coming to the deep learning part, we want to understand whether it is possible to train a neural network which, at the end, is able to recognize special Vogan diagrams. Thus, with the algorithm contained in `specialVoganDiagrams.sage`, we prepare the data as follows. For each connected Vogan diagram, we take the Cartan matrix of the corresponding complex simple Lie algebra and we add a row corresponding to the non-compact simple roots. Then we pad the matrix with 0's to get a $20 \times 20$ matrix. After that, we pair each Cartan matrix with a label: 0 if the corresponding Vogan diagram is special and 1 otherwise (`prepareData.sage`). Then we choose the proportion for test and training (`proportion.py`). Actually, if one does not want to prepare his own data, a ready-to-use data set is provided in the `data` folder. At this point, the data set is ready for the training. Using Keras with TensorFlow backend we define two neural networks: one is fully connected (`classificationSpecialDiagrams-fullyconnected.py`) while the other is convolutional (`classificationSpecialDiagrams-convolutional.py`)[1].

## `classificationSpecialDiagrams-fullyconnected.py`

After having loaded the data, the shapes of the training data and the test data are printed together with the numbers of special and non-special Vogan diagrams in the full data set. Then we build a fully connected neural network made of three `Dense` layers: the first and the last with 1000 nodes and the intermediate one with 300 nodes. All layers are spaced out by a `Dropout` layer. In the end, we put a `Dense` layer with softmax activation function.

We train the model with mini-batches of 100 elements and then we test the neural network on the set of testing data.

## `classificationSpecialDiagrams-convolutional.py`

After having loaded the data, the shapes of the training data and the test data are printed together with the numbers of special and non-special Vogan diagrams in the full data set. Then we build a convolutional neural network made of two `Conv1D` layers spaced out by

[1] The last two scripts shall be executed with Python3.

a `Maxpooling` layer. In the end, we put a fully connected neural network with two `Dense` layers spaced out by a `Dropout`, the last having softmax as activation function.

We train the model with mini-batches of 100 elements and then we test the neural network on the set of testing data.

### Observations

For both the neural networks, after the training one may notice that the predictions are very precise for non-special Vogan diagrams (with an error $< 1\%$), while they are less accurate for special Vogan diagrams. However, as special Vogan diagrams are far less than non-special ones, the total error remains small.

For what concerns predictions on Vogan diagrams of higher rank, it seems that both the two neural networks are not able to make accurate predictions, but this is still a work in progress and we will give updates about that very soon. Stay tuned!

# References

[1] A. Della Vedova and A. Gatti, *Almost Kaehler geometry of adjoint orbits of semisimple Lie groups*, arXiv:1811.06958v2 [math.DG].