Deciphering the Contents of Chemically-Trained Neural Networks into Physical Intuition

Eric Berquist

June 15th, 2017

Overview

Machine learning (ML) is seeing rapid growth in areas relevant to quantum chemistry, but how does it work?

- Topic: Are correct ML predictions in quantum chemistry *right for the right reasons*?
- Gap: We don't know if current approaches (ML architectures) will work more complex molecules or properties.
- Rationale: If a ML model is not right for the right reasons, there cannot be an expectation that it is transferable or extendable in any way.

We need to know if ML models are learning chemistry and not just numbers (von Neumann's elephant).

Overview

- The objective is to quantify what ML models trained on quantum chemical data are learning.
- The central hypothesis is that models are learning about molecular structure identically to how we apply chemical intuition.

This hypothesis will be tested by

- training neural networks (NNs) to replicate literature results,
- "seeing" what the currently-available models have learned using **relevance propagation**,
- attempt to predict more complex molecular properties than those found in the literature, and
- quantify if learning changes for more complex properties.

!!! Disclaimer !!!

The goal of this work is *not* to produce more accurate or more transferable models. The goal is to understand *how* and *why* models make (in)accurate predictions in terms of what they have learned.

What is machine learning?

Arthur Samuel (IBM), 1959: the subfield of computer science that gives

> *computers the ability to learn without being explicitly programmed.*

Tom Mitchell (CMU), 1997:

> *A computer program is said to learn from experience E with respect 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.*

Machine learning will solve all our problems

Harvard Business Review

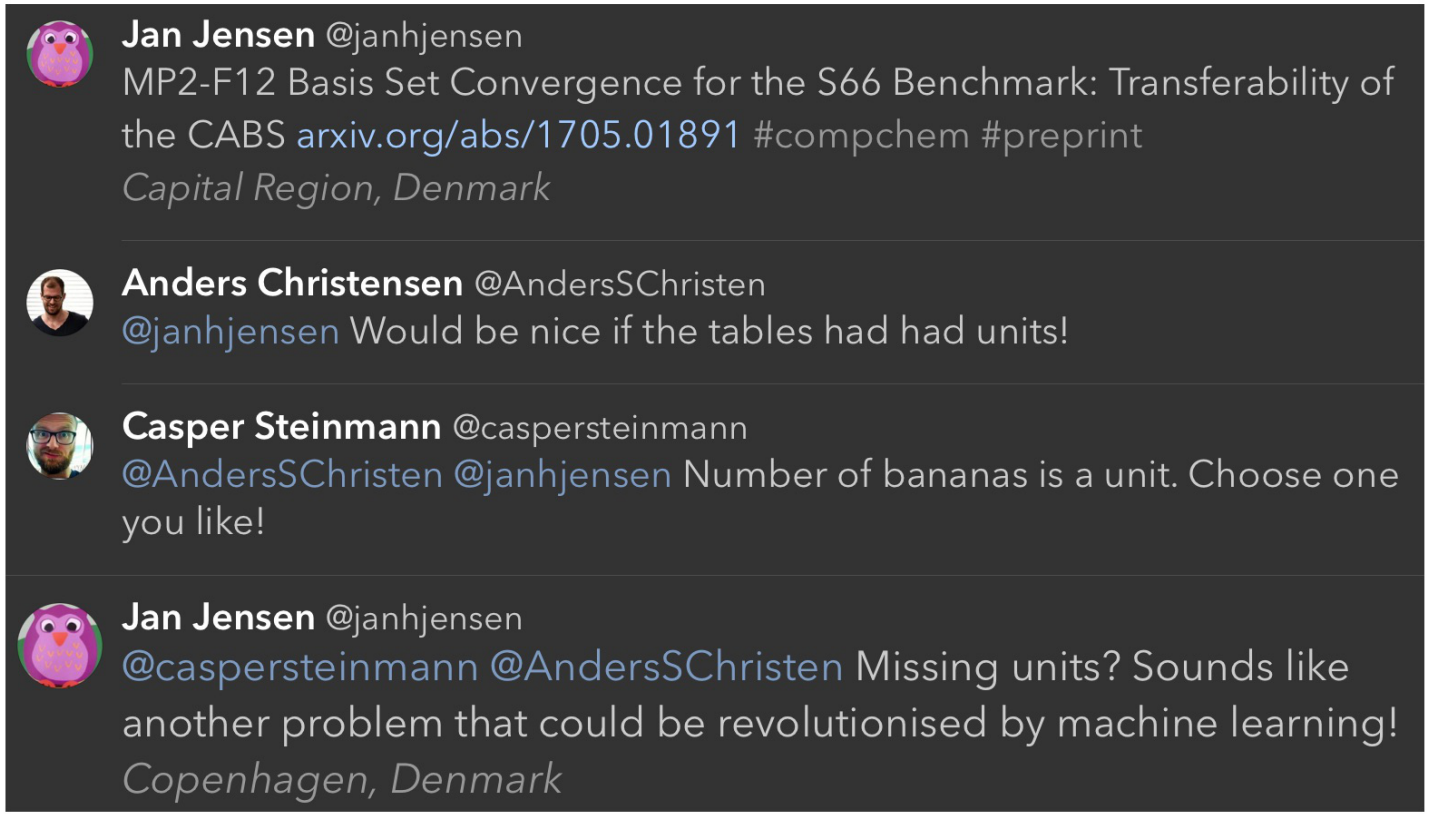
ANALYTICS

A Guide to Solving Social Problems with Machine Learning

by Jon Kleinberg, Jens Ludwig, and Sendhil Mullainathan

DECEMBER 08, 2016

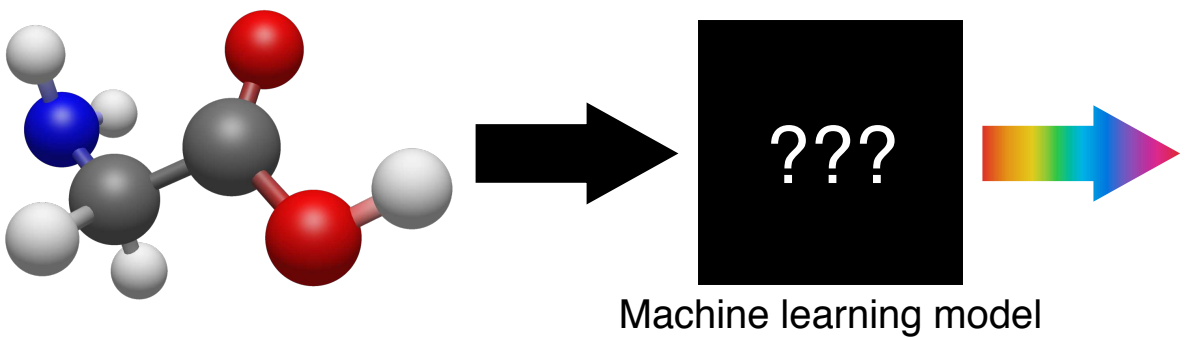
Machine learning will solve all our problems




THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS

Machine learning has a perception problem

Machine learning is a "fad" and produces all these great results, but we joke semi-seriously that we don't know what's going on under the hood, even though it will solve all our problems.



Objective

- Peek inside the black box and see if ML models are "learning chemistry".

Rationale

- Building complex ML models that can do real, useful chemistry in a *general* manner is impossible without proving meaningfulness of simpler models.
- Clearly the dozen or so papers from 2016-2017 show that accurate predictions can be made even under the assumption of black-box models.
- Additionally, if we can interpret the model directly, then perhaps eventually we can interpret chemistry using the model itself and not just predictions.

Rationale

Is it alright to accept the use of NNs that are not truly transferable (B3LYP, M06)? Maybe this works for prediction results, but we will repeat the history of DFT.

Obituary: Density Functional Theory (1927–1993)

Peter M. W. Gill

School of Chemistry, University of Nottingham, Nottingham NG7 2RD, United Kingdom.
Manuscript received: 7 March 2002 (e-mail: Peter.Gill@nottingham.ac.uk).
Final version: 14 March 2002.

REPORT

THEORETICAL CHEMISTRY

Density functional theory is straying from the path toward the exact functional

Michael G. Medvedev,[1,2,3]*† Ivan S. Bushmarinov,[1]*† Jianwei Sun,[4]‡ John P. Perdew,[4,5]† Konstantin A. Lyssenko[1]†

The theorems at the core of density functional theory (DFT) state that the energy of a many-electron system in its ground state is fully defined by its electron density distribution. This connection is made via the exact functional for the energy, which minimizes at the exact density. For years, DFT development focused on energies, implicitly assuming that functionals producing better energies become better approximations of the exact functional. We examined the other side of the coin: the energy-minimizing electron densities for atomic species, as produced by 128 historical and modern DFT functionals. We found that these densities became closer to the exact ones, reflecting theoretical advances, until the early 2000s, when this trend was reversed by unconstrained functionals sacrificing physical rigor for the flexibility of empirical fitting.

Transferability

Literature usage:

- No need for reparametrization from system to system
- Limited to organic molecules, train small (9 heavy atoms), test larger (10 heavy atoms)
- Charge and spin: neutral and closed-shell singlet

A better definition in terms of examples:

- Does the same model work for optimized and non-equilibrium (MD) structures?
- Does the model work for charged systems?
- Does the model work for systems with unpaired electrons?
- Does the model work for *excited states*?

Specific Aims

1. Reproduce existing neural network models for molecular properties from the literature.
2. Characterize the parameters learned by existing neural network models from the literature using relevance propagation.
3. Train supervised neural networks on complex molecular properties.
4. Characterize the parameters learned for complex molecular properties using relevance propagation and unsupervised neural networks.

Background

Introduction to machine learning

Supervised learning:

- Learn to predict an output given an input

Unsupervised learning:

- Discover a good internal representation of the input
- Learn to reconstruct the input from itself *non-trivially*

Introduction to machine learning

Classification:

- Given a set of data, identify the classes that the data belongs to
- Predict what group a piece of data is a member of
- Output: Discrete, categorical
- Example: x could be a cat, dog, or bird, and is a bird $\rightarrow y = (0, 0, 1)$

Regression:

- Given a set of data, find the best relationship that represents the set of data
- Output: Continuous, numerical
- Example: Find m and b in $y = mx + b$

Simplest form: univariate linear regression

Hypothesis:

$$h_\theta(x) = \theta_0 + \theta_1 x$$

Parameters:

$$\theta_0, \theta_1$$

Cost/penalty function
($m = \#$ of training inputs, $y =$ exact prediction):

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

Goal:

$$\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$$

Parameter optimization

Finding the set of coefficients that minimize the cost function:

$$\frac{\partial J}{\partial \theta_j} \stackrel{!}{=} 0$$

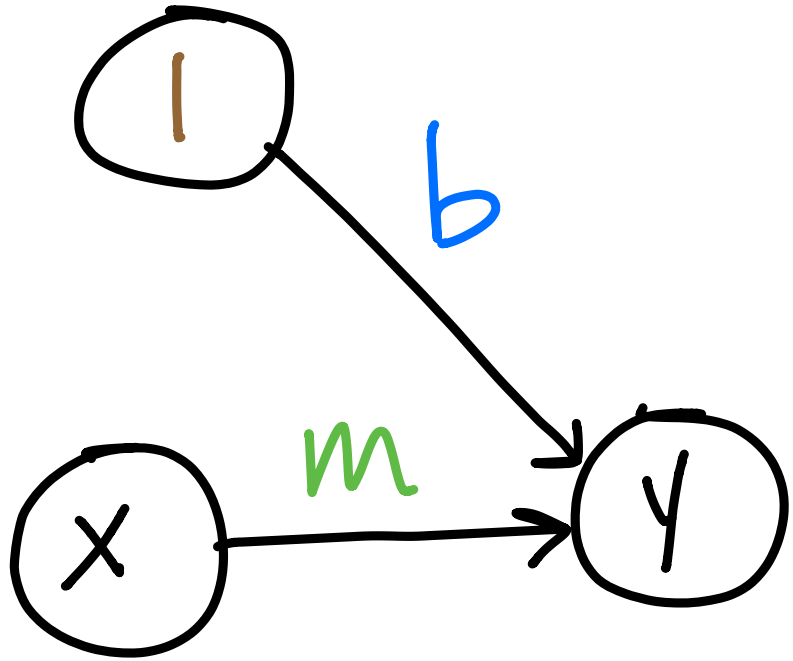
$$= \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right) x^{(i)}$$

which are used in gradient descent (or an equivalent) algorithm:

1: **repeat**
2: $\quad \theta_0 \leftarrow \theta_0 - \eta \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$
3: $\quad \theta_1 \leftarrow \theta_1 - \eta \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$
4: **until** convergence

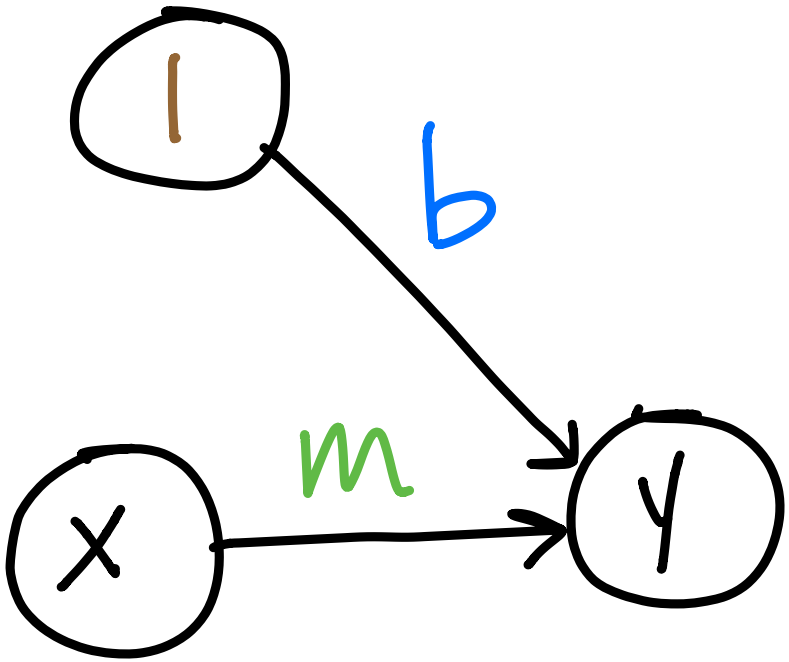
where η is the learning rate (a hyperparameter).

Linear regression using a neural network



$$y = (m * x) + (b * 1)$$

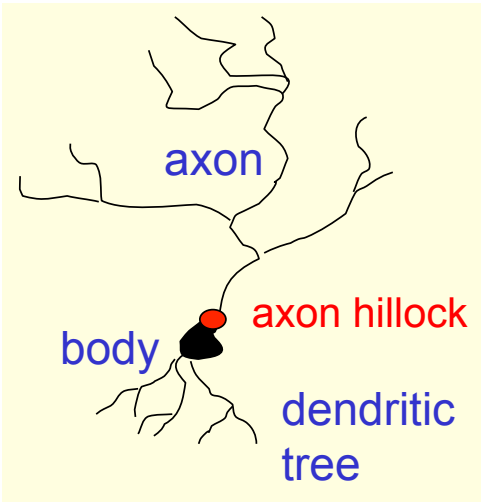
Linear regression using a neural network

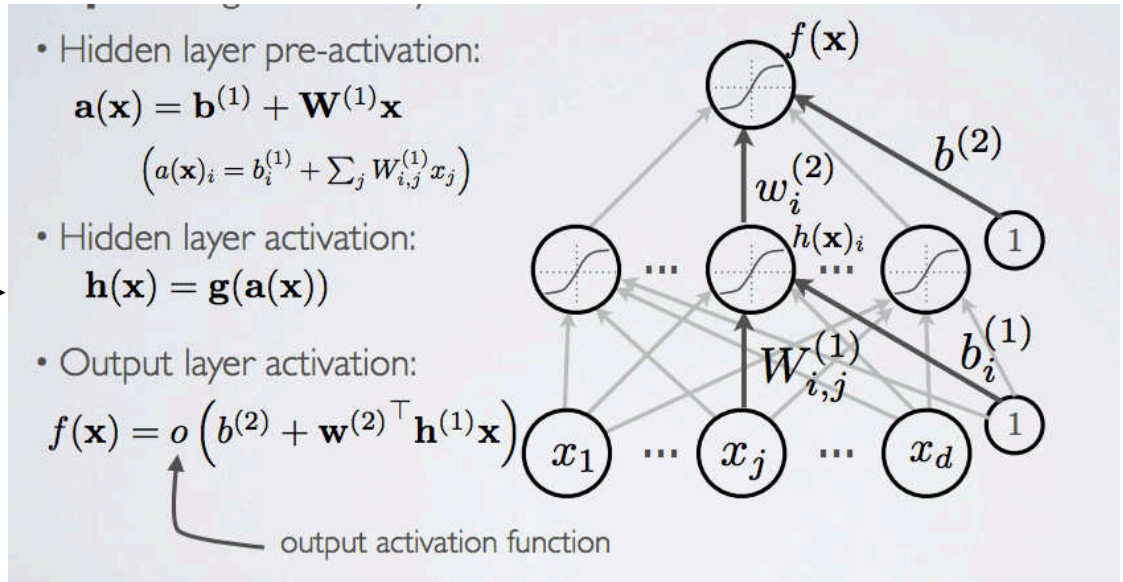


$$a = (m * x) + (b * 1)$$
$$y = g(a)$$
$$g(z) = z$$

$$y = (m * x) + (b * 1)$$

General architecture of neural networks



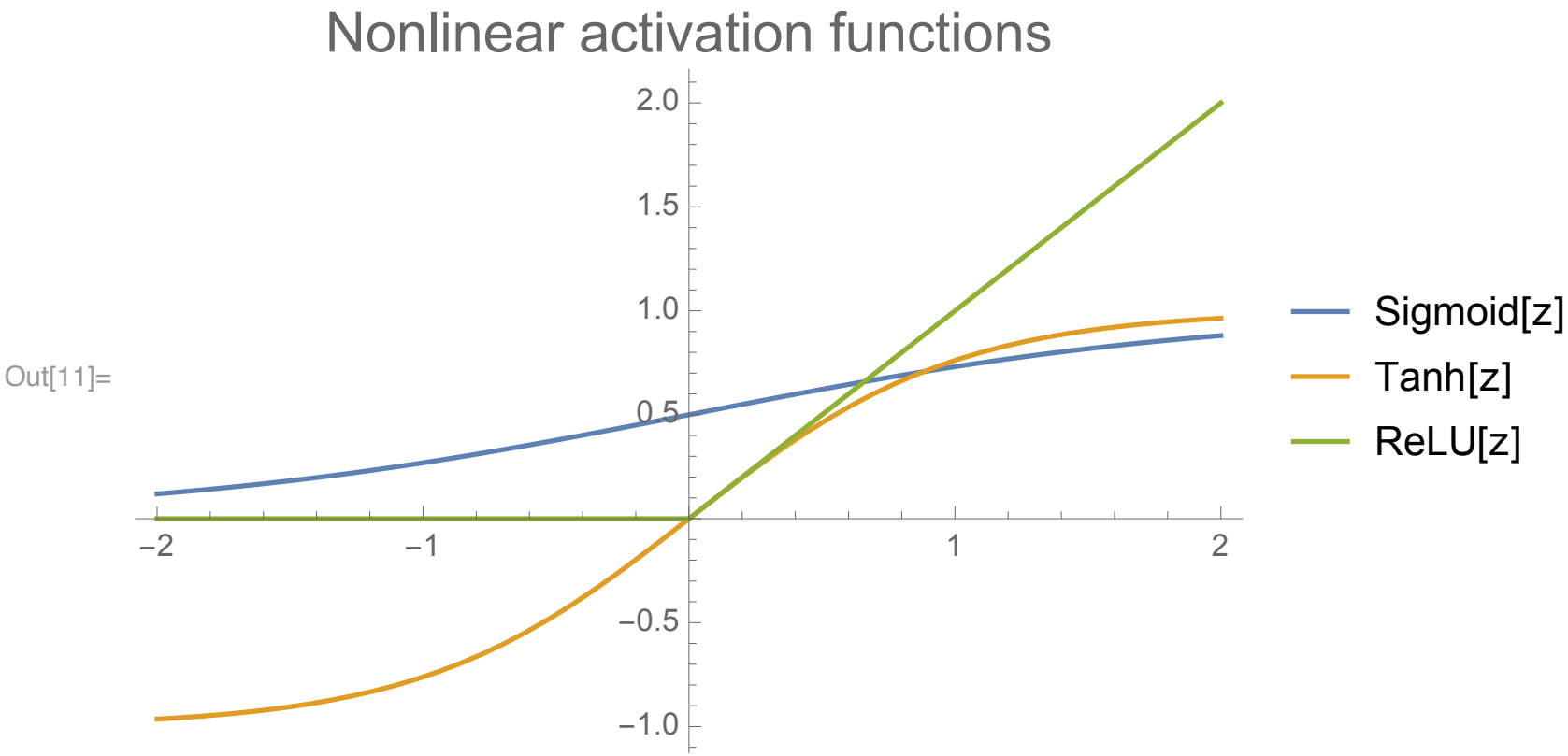


Brain

Model Architecture

Read from left $\rightarrow$ right or bottom $\rightarrow$ top.

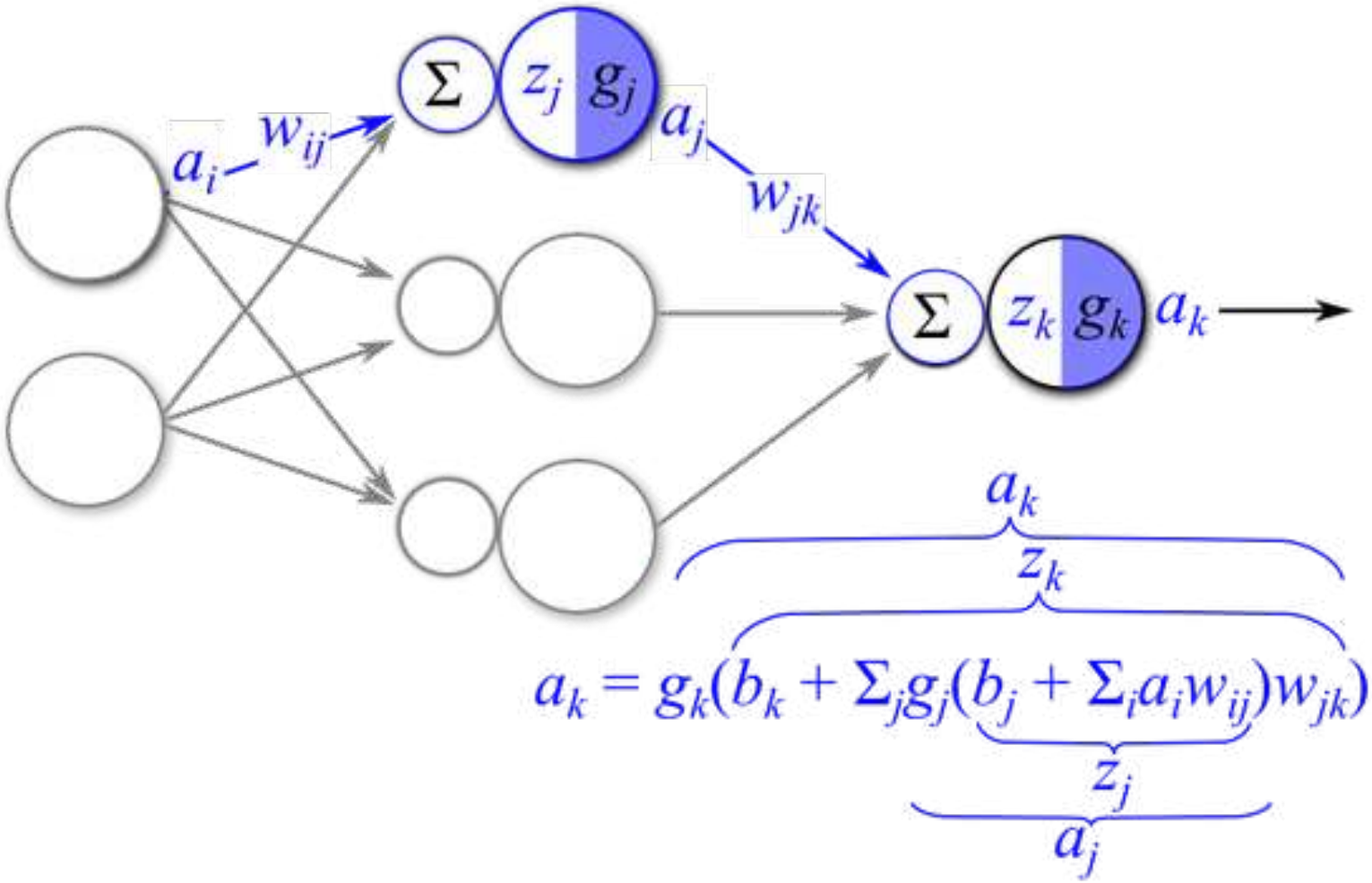
Neural networks perform nonlinear transformations



The combination of hidden layers and nonlinear connections between layers makes them *universal function approximators*.

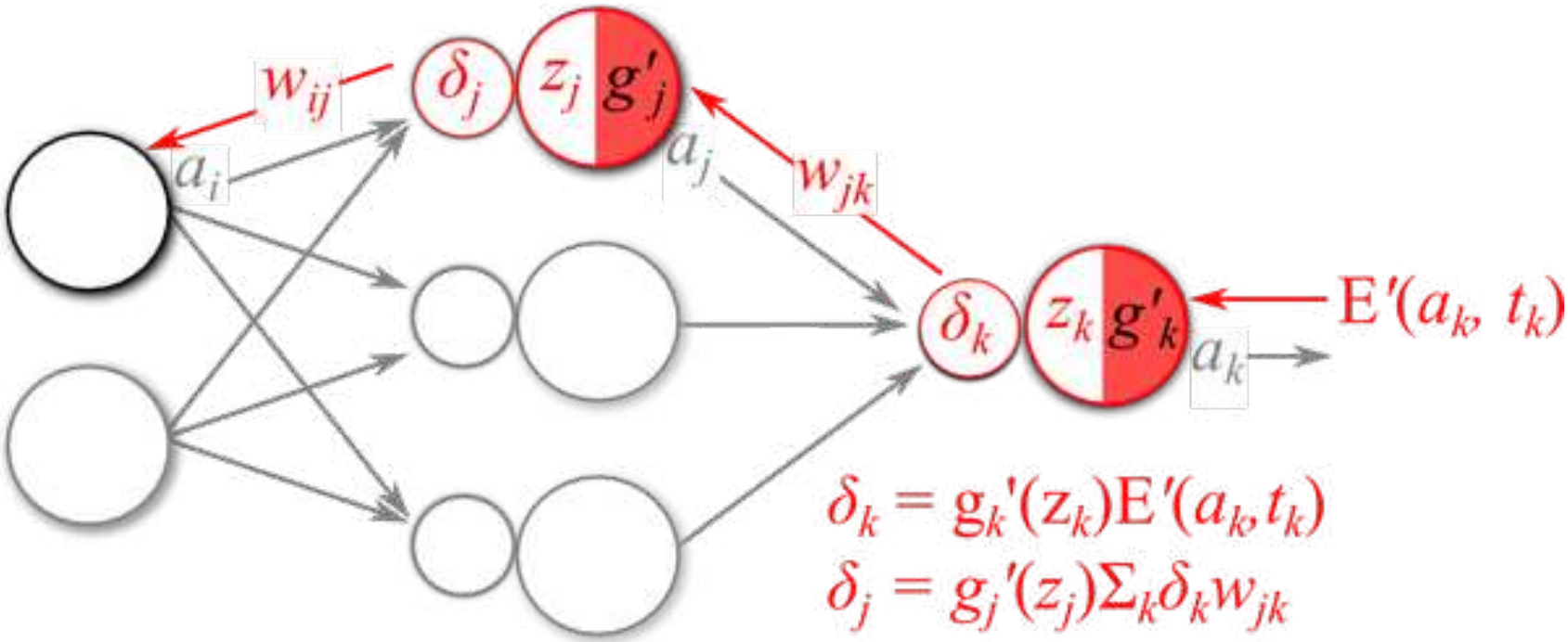
Parameter optimization: training neural networks

Step 1. Forward propagation of the input signals



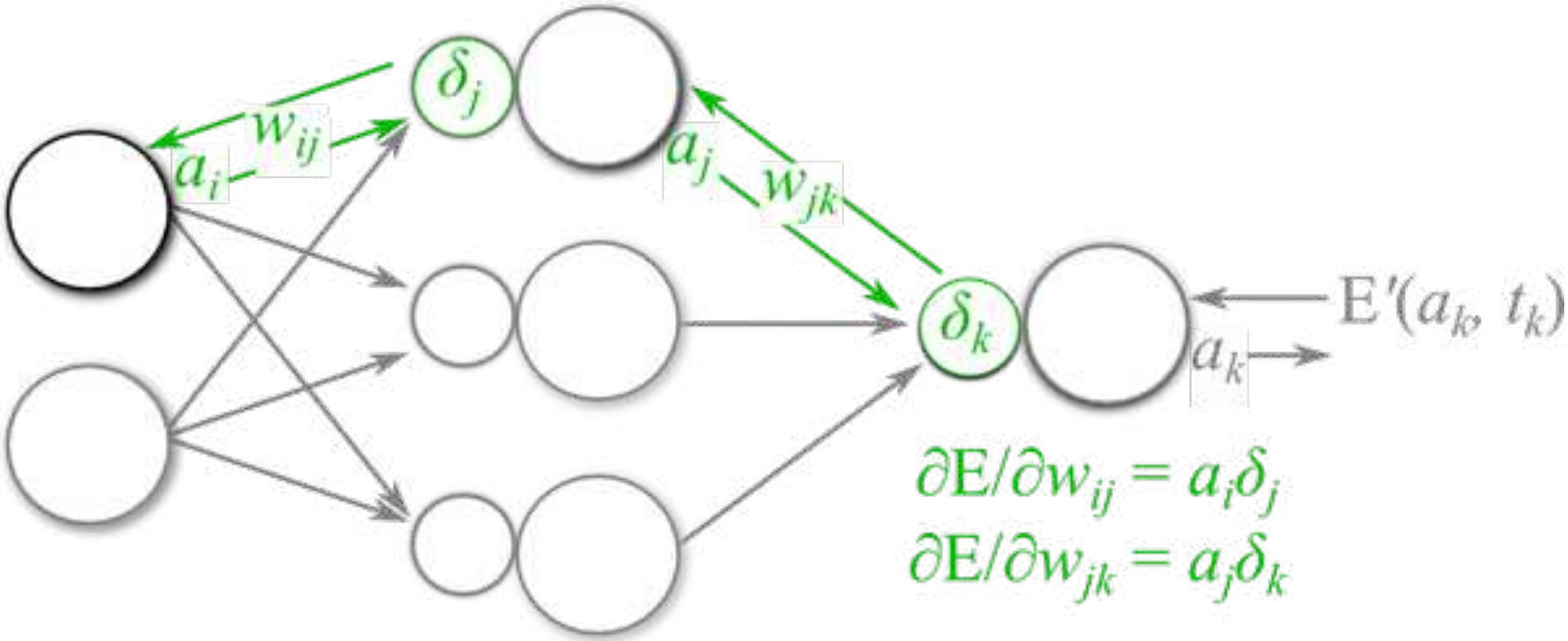
Parameter optimization: training neural networks

Step 2. Back propagation of the error signals



Parameter optimization: training neural networks

Step 3. Calculate parameter gradients from error signals and activations



Step 4. Update parameters from gradients

$$w_{ij} = w_{ij} - \eta(\partial E/\partial w_{ij})$$
$$w_{jk} = w_{jk} - \eta(\partial E/\partial w_{jk})$$

for learning rate η

An (imperfect) connection between neural networks and quantum chemistry

- The fundamental components of the network (kind of neuron activation functions, convolution or direct connection) are like the Hamiltonian, and
- the number of components in each network layer, the number of layers, and the input representation are like the size and type of basis set.

Increasing the number of layers and number of nodes per layer is like lowering the variational bound of the network, and weights play a similar role to MO coefficients.

The connection between basis sets and input featurization

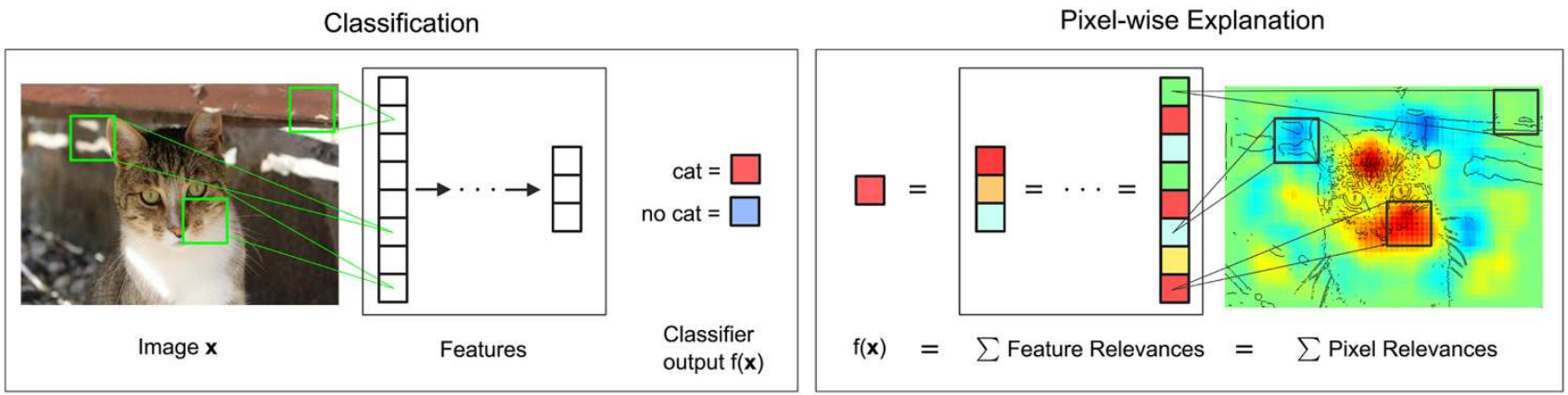
What form the input of any ML model takes plays a large role on how well it performs.

- Adding diffuse functions to a basis set enables finding the correct (qualitative) answers for anions.
- Adding better input features (molecular descriptors) enables the model architecture to find better weights, leading to more accurate predictions.

Layer-wise Relevance Propagation (LRP)

Current literature: when identifying the primary contents of an image (classification), what pixels were strong evidence for its classification, and what pixels indicated that it may be something else (evidence against)?

- *Not* what pixels were unimportant



Red is evidence for the classification choice, blue is evidence against.

LRP example for image classification

Classification Result

Class	Prediction Score
tabby, tabby cat	0.7864
tiger cat	0.1064
Egyptian cat	0.0981
lynx, catamount	0.0081
cougar, puma, catamount, mountain lion, painter, panther, Felis concolor	0.0004
coyote, prairie wolf, brush wolf, Canis latrans	0.0001
timber wolf, grey wolf, gray wolf, Canis lupus	0.0001
tiger, Panthera tigris	0.0001
Persian cat	0.0001
grey fox, gray fox, Urocyon cinereoargenteus	0.0000

http://www.heatmapping.org/caffe.html

LRP example for image classification

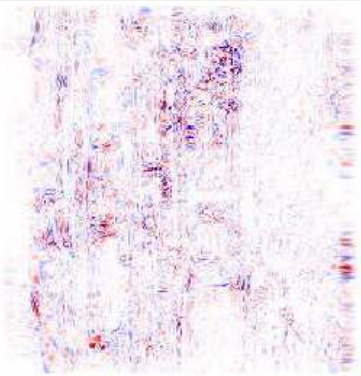
Classification Result

Class	Prediction Score
monastery	0.2814
bell cote, bell cot	0.2120
castle	0.2026
palace	0.0590
barn	0.0456
picket fence, paling	0.0245
boathouse	0.0228
worm fence, snake fence, snake-rail fence, Virginia fence	0.0224
suspension bridge	0.0222
stupa, tope	0.0173

LRP example for image classification

Classification Result

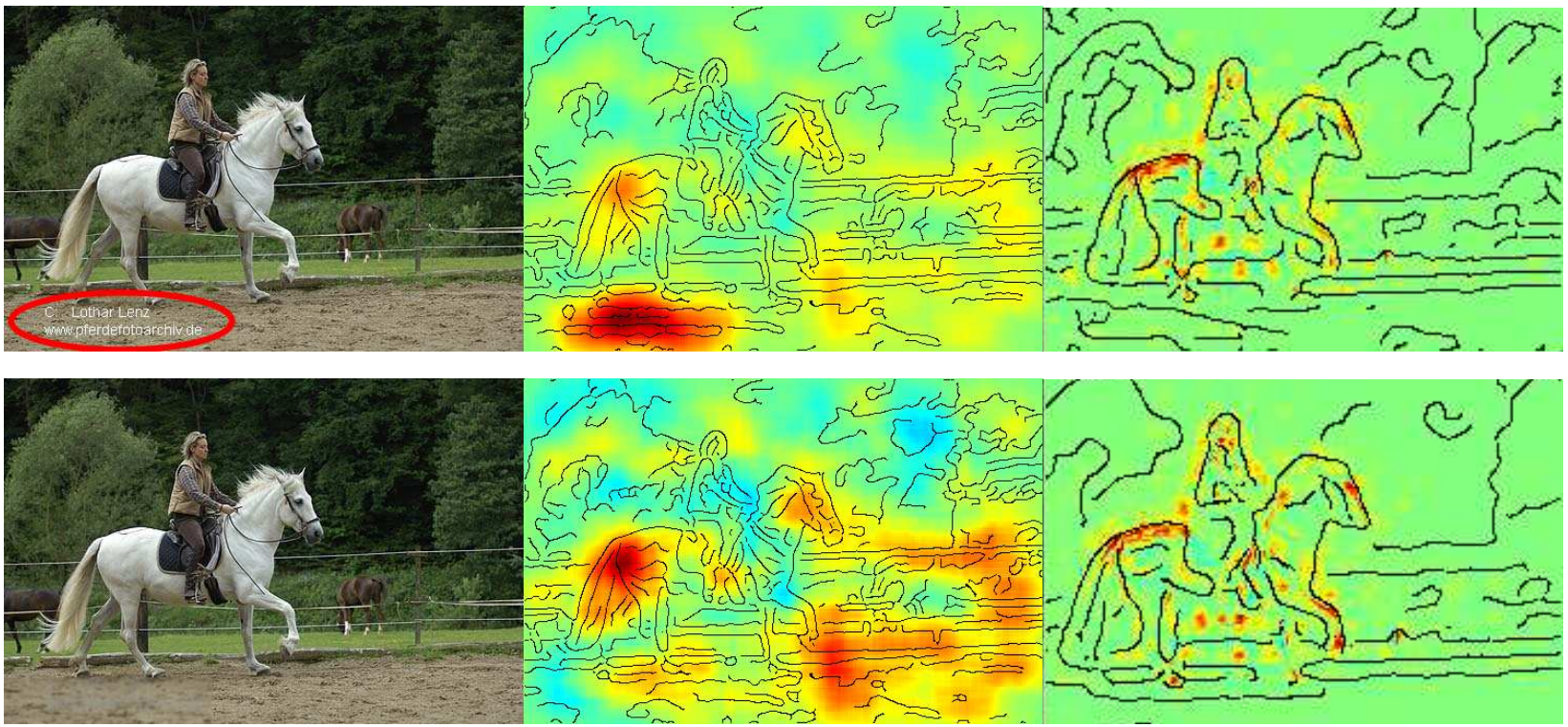
Class	Prediction Score
forklift	0.8432
bassoon	0.0797
padlock	0.0125
ski	0.0103
accordion, piano accordion, squeeze box	0.0061
oboe, hautboy, hautbois	0.0054
fountain pen	0.0043
paintbrush	0.0028
sax, saxophone	0.0026
screwdriver	0.0020



We learn something about what the model learned, even when classification fails.

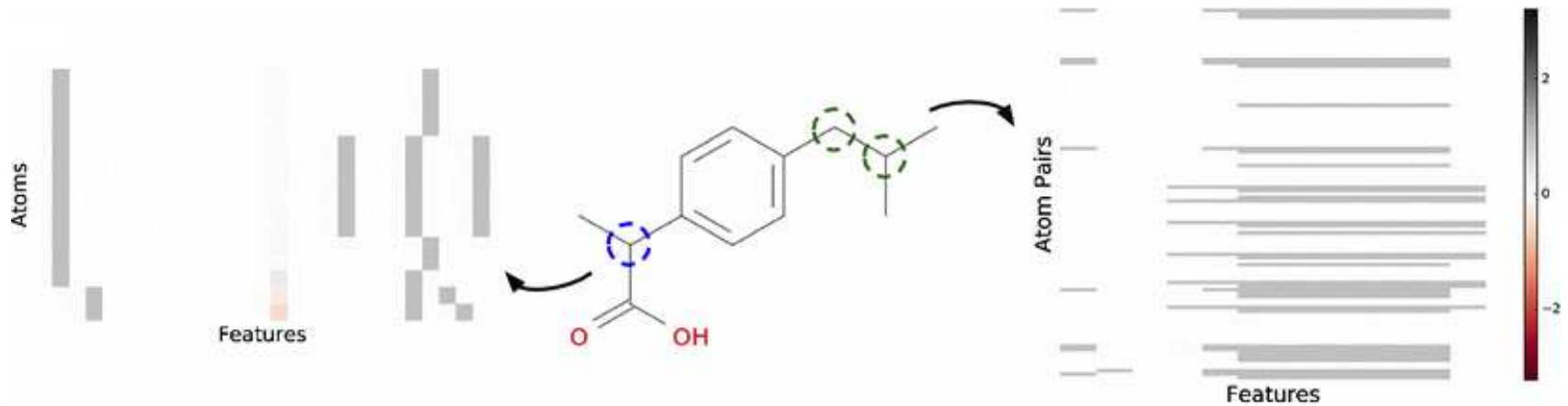
Learning the right thing for the right reason

Image | FV | DNN



average heatmaps

LRP for quantum chemistry in machine learning



This work will show whether models are "learning the caption" or "learning the head and tail".

Specific Aims

Aim #1: Reproduction of Existing Literature Neural Networks

O. Anatole von Lilienfeld has done the most work for predicting molecular properties using several different ML architectures

- "Fast machine learning models of electronic and energetic properties consistently reach approximation errors better than DFT accuracy", arxiv.org/abs/1702.05532
- Use the Graph Convolutions neural network architecture combined with the Molecular Graph input (MG/GC)

Molecular properties:

- Zero-point (vibrational) energy: $E_{\mathrm{ZPVE}} = \frac{1}{2} h \sum_i^{\text{normal modes}} \nu_i$
- Isotropic polarizability (static, $\omega = 0$): $\alpha_{\mathrm{iso}} = \bar{\alpha} \equiv \frac{1}{3}(\alpha_{xx} + \alpha_{yy} + \alpha_{zz})$

Justification for aim #1

Why choose the GC architecture over DTNN, ANAKIN-ME, ...?

- Want comparison against literature results (more on this later), these so far are molecular energies only.
- Where is the code?

Why *not* look at molecular energies?

- ML needs to be capable of spectroscopy, calculations for which are much more expensive than energies/trajectories.
- Comparison against experiment

Training inputs: the QM9 molecular database

We report computed geometric, energetic, electronic, and thermodynamic properties for 134k stable small organic molecules made up of CHONF. These molecules correspond to the subset of all 133,885 species with up to nine heavy atoms (CONF) out of the GDB-17 chemical universe of 166 billion organic molecules. We report geometries minimal in energy, corresponding harmonic frequencies, dipole moments, polarizabilities, along with energies, enthalpies, and free energies of atomization. All properties were calculated at the B3LYP/6-31G(2df,p) level of quantum chemistry.

Molecular Graph input representation: single atom features

Feature	Description	Size
Atom type	H, C, N, O, or F (one-hot)	5
Chirality	R or S (one-hot or null)	2
Formal charge	Integer electronic charge	1
Partial charge	Calculated partial charge	1
Ring sizes	For each ring size (3-8), the number of rings that include this atom	6
Hybridization	sp, sp^2, or sp^3 (one-hot or null)	3
Hydrogen bonding	Whether this atom is a hydrogen bond donor and/or acceptor (binary values)	2
Aromaticity	Whether this atom is part of an aromatic system	1
		21

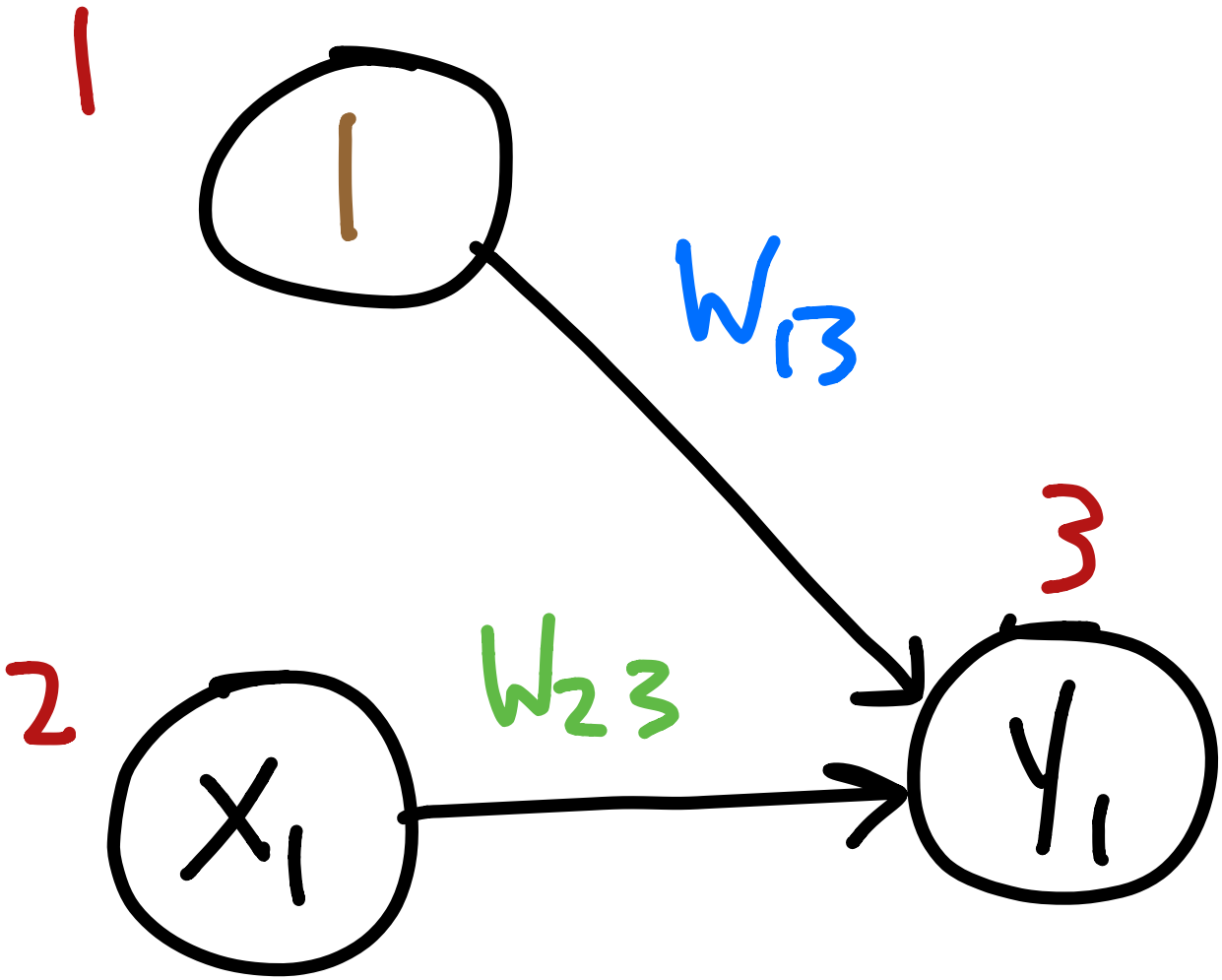
Molecular Graph input representation: atom pair features

Feature	Description	Size
Bond type	Single, double, triple, or aromatic (one-hot or null)	4
Graph distance	For each distance (1-7), whether the shortest path between the atoms in the pair is less than or equal to that number of bonds (binary values)	7
Same ring	Whether the atoms in the pair are in the same ring	1
Spatial distance	The Euclidean distance between the two atoms	1
		13

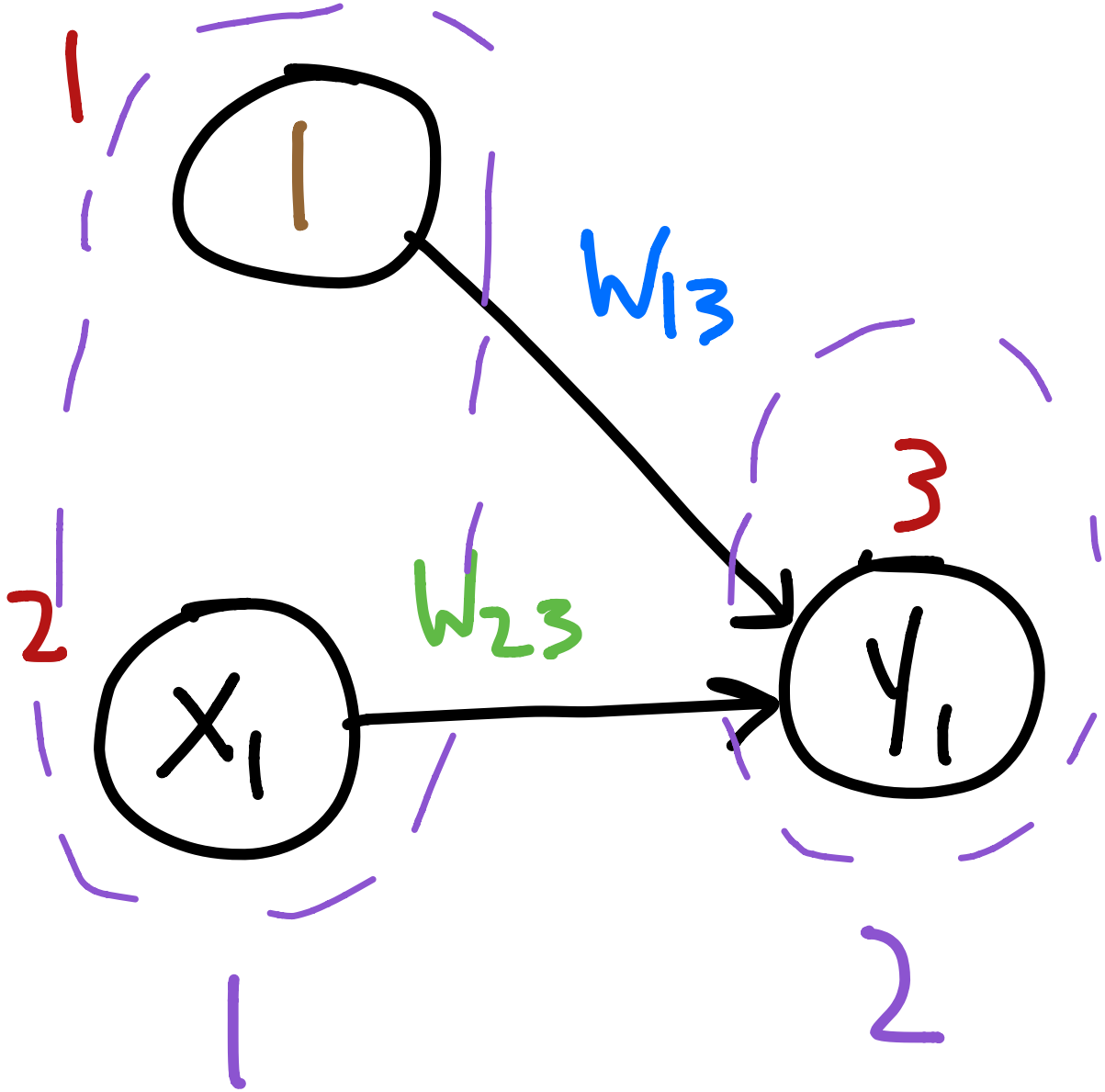
Aim #2: Characterization of Existing Literature Neural Networks

- Transfer the layer-wise relevance propagation (LRP) technique from image classification to a regression problem.
- The expected outcome is a clear connection between the input molecular representations and predicted outputs.
- If there are no connections, we still learn a substantial amount about what the neural networks have learned.

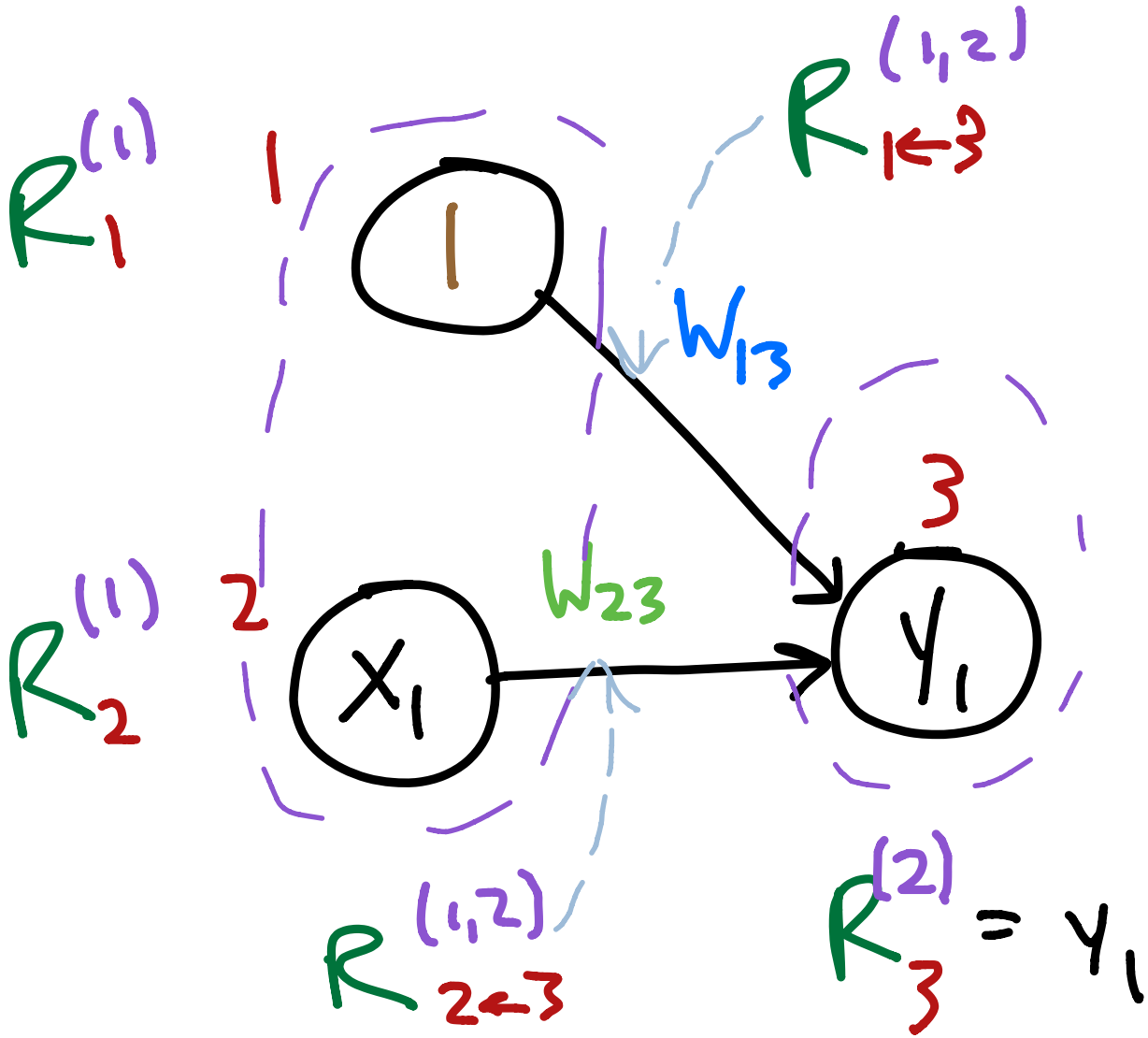
LRP concrete example



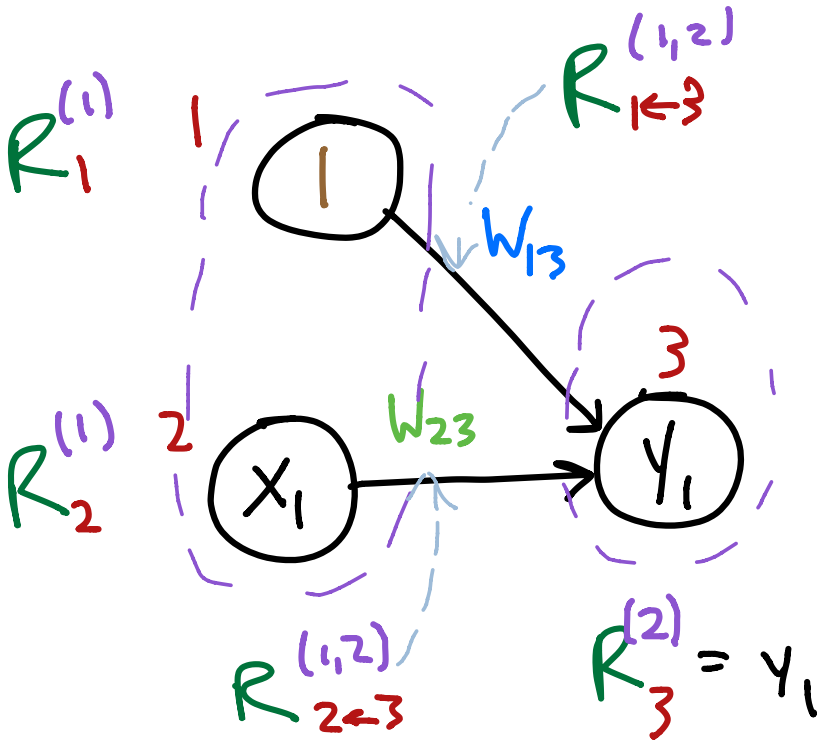
LRP concrete example



LRP concrete example



LRP concrete example



Relevance of a single message:

$$R_{i\leftarrow k}^{(l,l+1)} = R_k^{(l+1)} \frac{a_i w_{ik}}{\sum_h a_h w_{hk}}$$

$$R_{2\leftarrow 3}^{(1,2)} = R_3^{(2)} \frac{a_{23} w_{23}}{a_{23} w_{23}}$$

Relevance of an individual node is the sum of all incoming messages:

$$R_2^{(1)} = R_{2\leftarrow 3}^{(1,2)}$$

Total relevance per layer is the sum of relevances of individual nodes in layer:

$$R^{(1)} = R_1^{(1)} + R_2^{(1)}$$

$$R^{(2)} = R_3^{(2)}$$

Relevance is conserved layer-by-layer:

$$R^{(1)} = R^{(2)} = f(x) = y_1$$

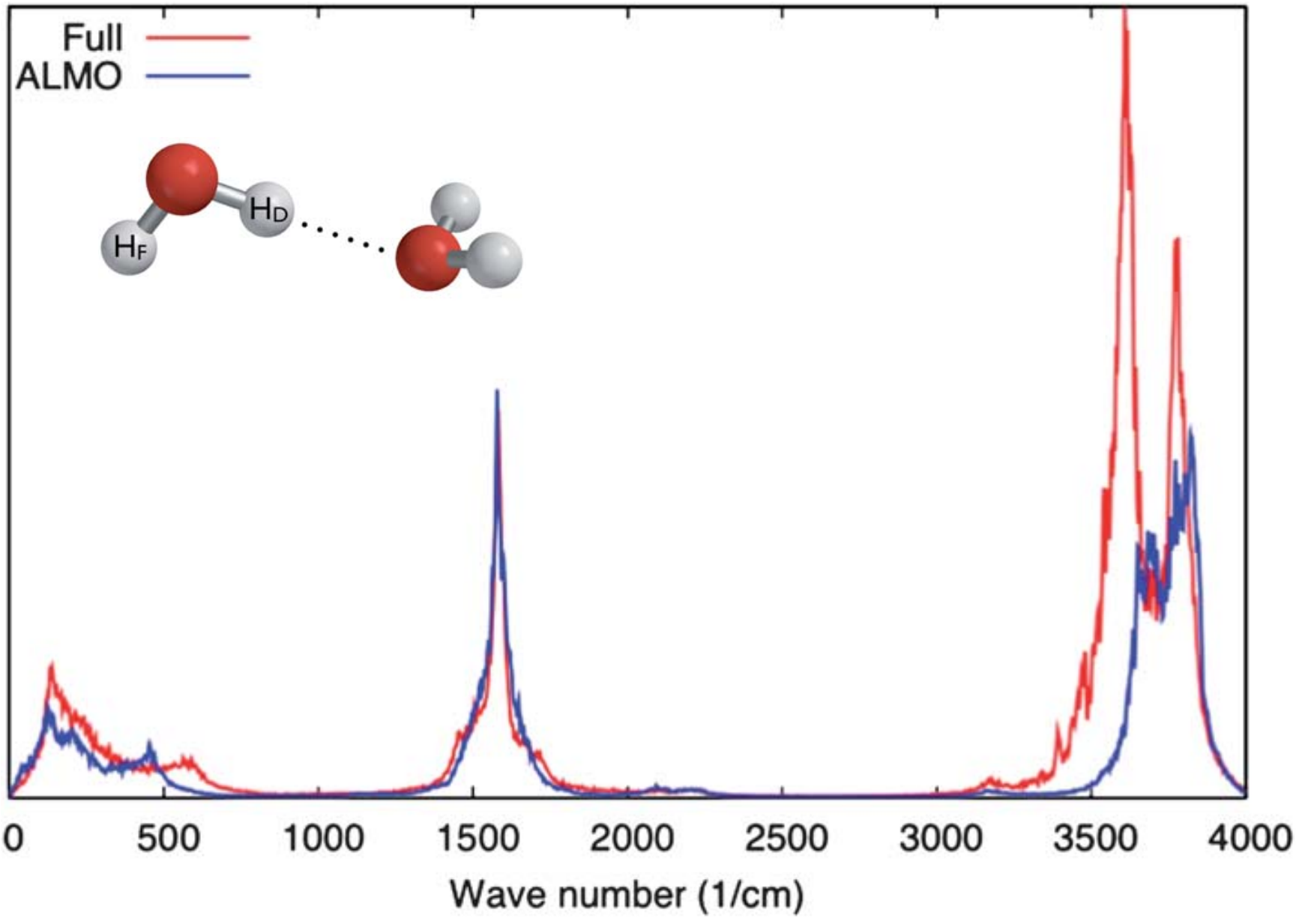
What LRP is *not* doing

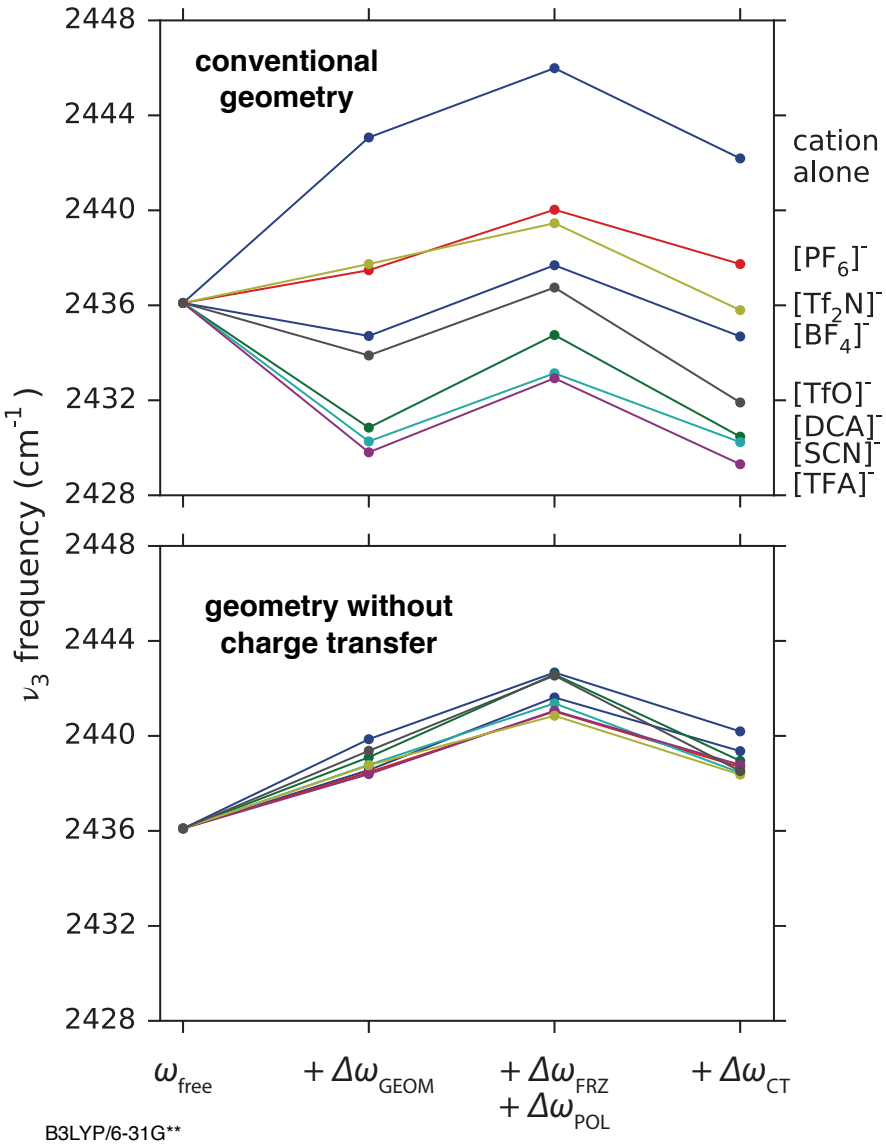
- This is *not* a direct inspection of what the NN has learned!
- Looking directly at NN weights is similar to looking at MO coefficients. Once the number of them grows, the "importance" of a single one diminishes greatly, and the number of nodes/weights grows even quicker than the number of MO coefficients for a reasonable quantum chemical calculation. The ability for direct inspection becomes impossible.

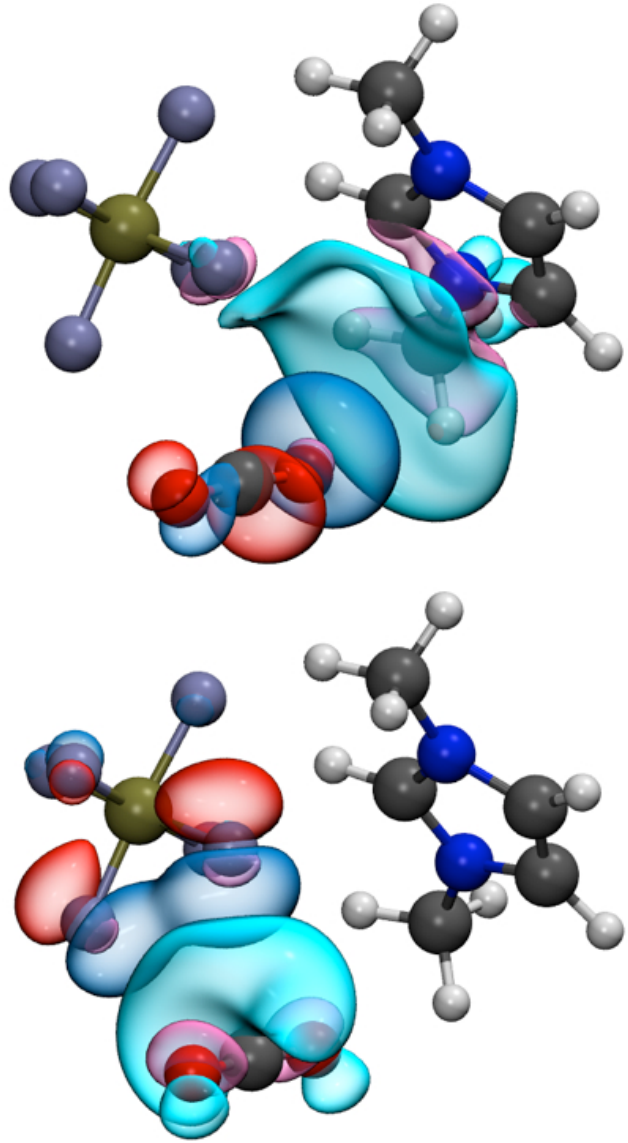
A better analogy

The use of layer-wise relevance propagation is identical to interaction energy analysis.

- SAPT and ALMO-EDA give the best theoretical, physically-intuitive, and quantitative insight into how molecules interact by *decomposing* the interaction energy.
- LRP gives a *decomposition* of the predicted output in terms of the input features.


Full
ALMO
H_D
H_F
0
500
1000
1500
2000
2500
3000
3500
4000
Wave number (1/cm)





COVPs depicting charge donation from CO_2 into the cation and from the anion into CO_2; ionic liquid is $[C_1C_1im^+][PF_6^-]$

Aim #3: Training Neural Networks for Complex Molecular Properties

Train the same architecture from aim #1 (MG/GC) on QM9 for more complex molecular properties:

- Parallel 1st hyperpolarizability (static, $\omega_a = \omega_b = 0$): $\beta_{\parallel} \equiv \frac{3}{5|\mu|} \sum_{i,j=x,y,z} \beta_{iij} \mu_j$
- All vibrational frequencies: $\{\tilde{\nu}\}_{\text{normal modes}}$

Aim #4: Characterization of Novel Neural Networks

Start by applying the same analysis techniques from aim #2 to the trained networks from aim #3.

- The expected outcome is that similar features show similar relevance patterns for $\bar{\alpha}/\beta_{\|}$ and $E_{\mathrm{ZPVE}}/\{\tilde{\nu}\}$.
- If not, either the size or the fundamental components of the network architecture are insufficient for describing more complex molecular properties.
- Combined with systematic changes to input features, LRP can still show *where* the networks are deficient.

Aim #4: Characterization of Novel Neural Networks

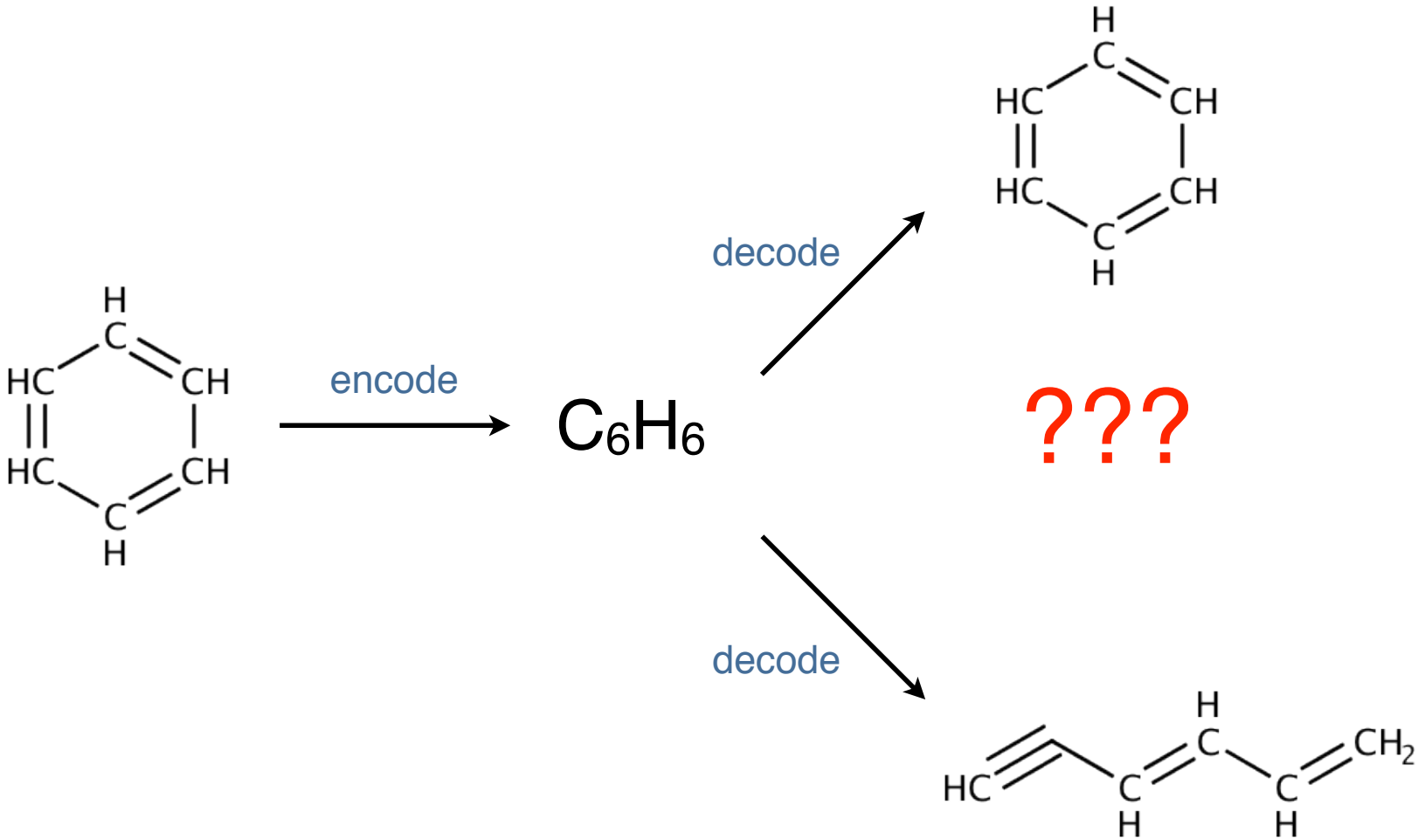
The original authors of Graph Convolutions found that using a smaller set of features than those in the full Molecular Graph representation still agreed quantitatively for classification.

- Atom type, bond type, and graph distance

Unsupervised learning with an autoencoder can automatically find if these or other input features are important.

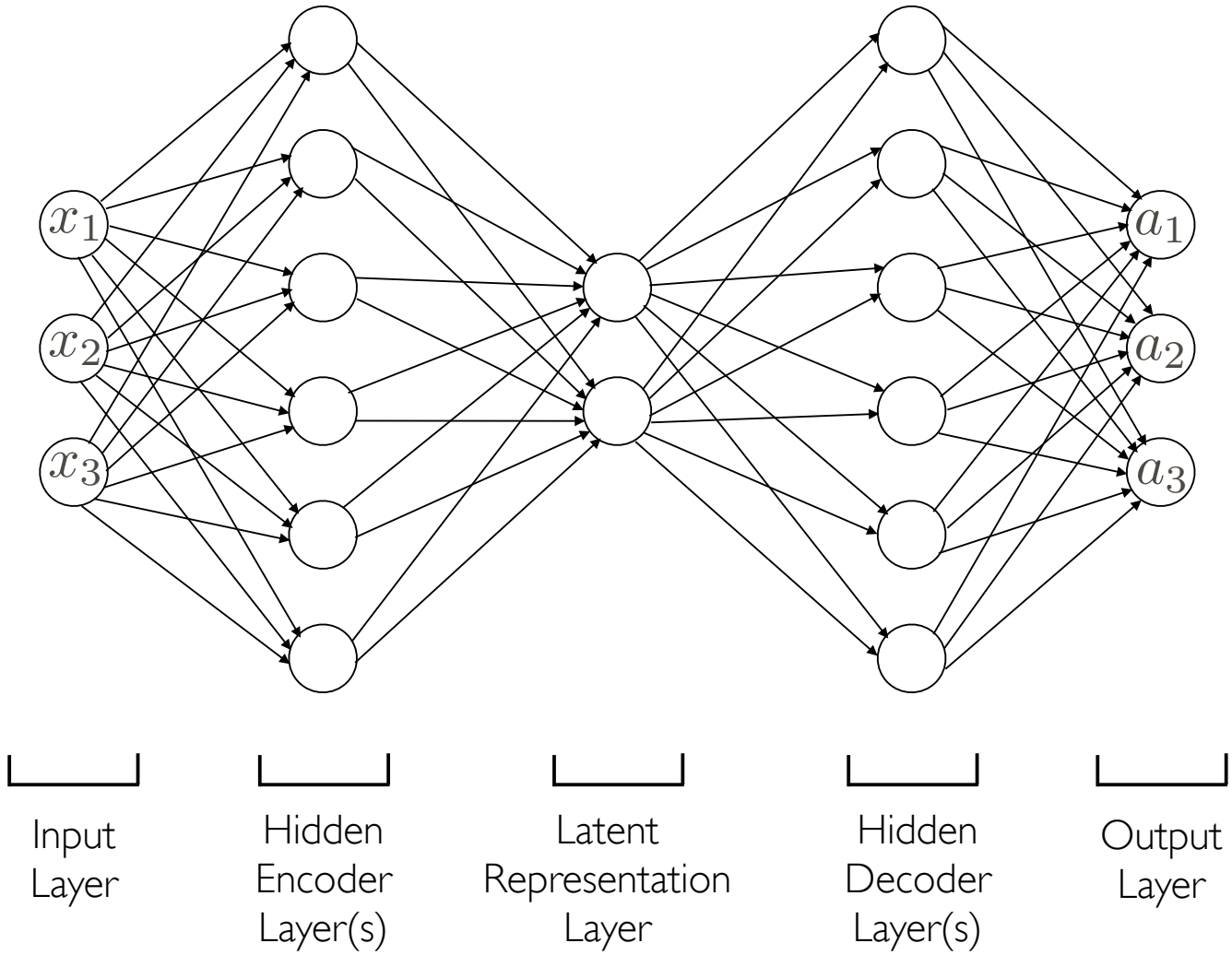
- The expected outcome is that adding an unsupervised learning stage provides a reduced-dimensionality molecular representation connecting directly to chemical intuition
- If not, many aspects of the autoencoder's architecture can be systematically changed.

Learning a compressed representation



Attempt to reduce some complex input into a smaller form (code) that can accurately be turned back into the complex input

Autoencoder structure



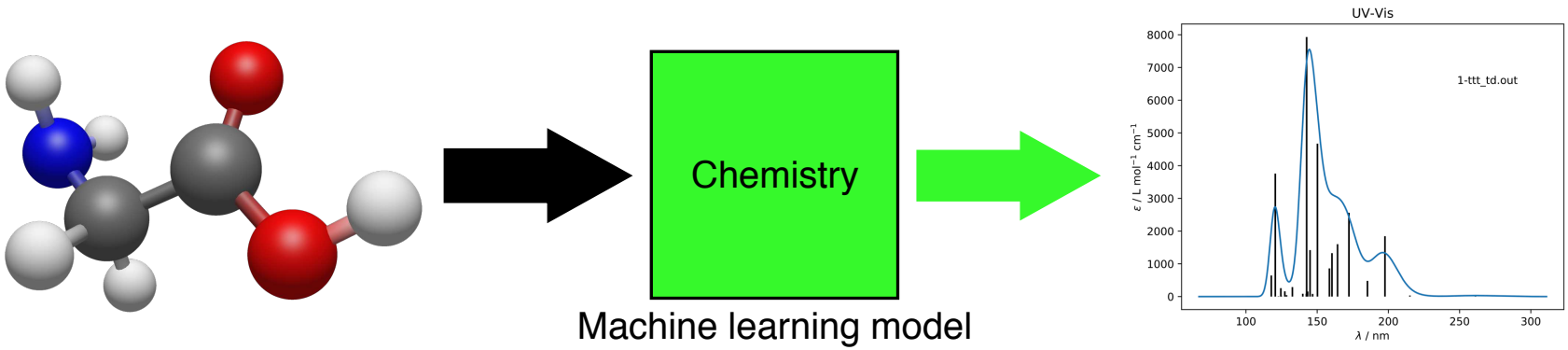
A *denoising* autoencoder adds noise to the input during training.

Approximate Timeline

Specific Aim	Task	# of Months
1	code development: forming pipeline	3
1	model training	3
2	code development: adapt LRP to pipeline	3
2	analysis development	3
3	hyperpolarizability calculations	2
3	model training	3
4	code development: DAE	3
4	model training	3
4	analysis	3
Total		26

Significance

- This is the first attempt at understanding the parameters of ML models used to predict microscopic and macroscopic molecular properties, rather than treating the models as black boxes that cannot be understood.
- This is the first use of relevance propagation outside of image classification and for any regression technique.
- The models trained here are the first proof-of-concept ML predictions of higher-order nonlinear optical properties and vibrational spectra.

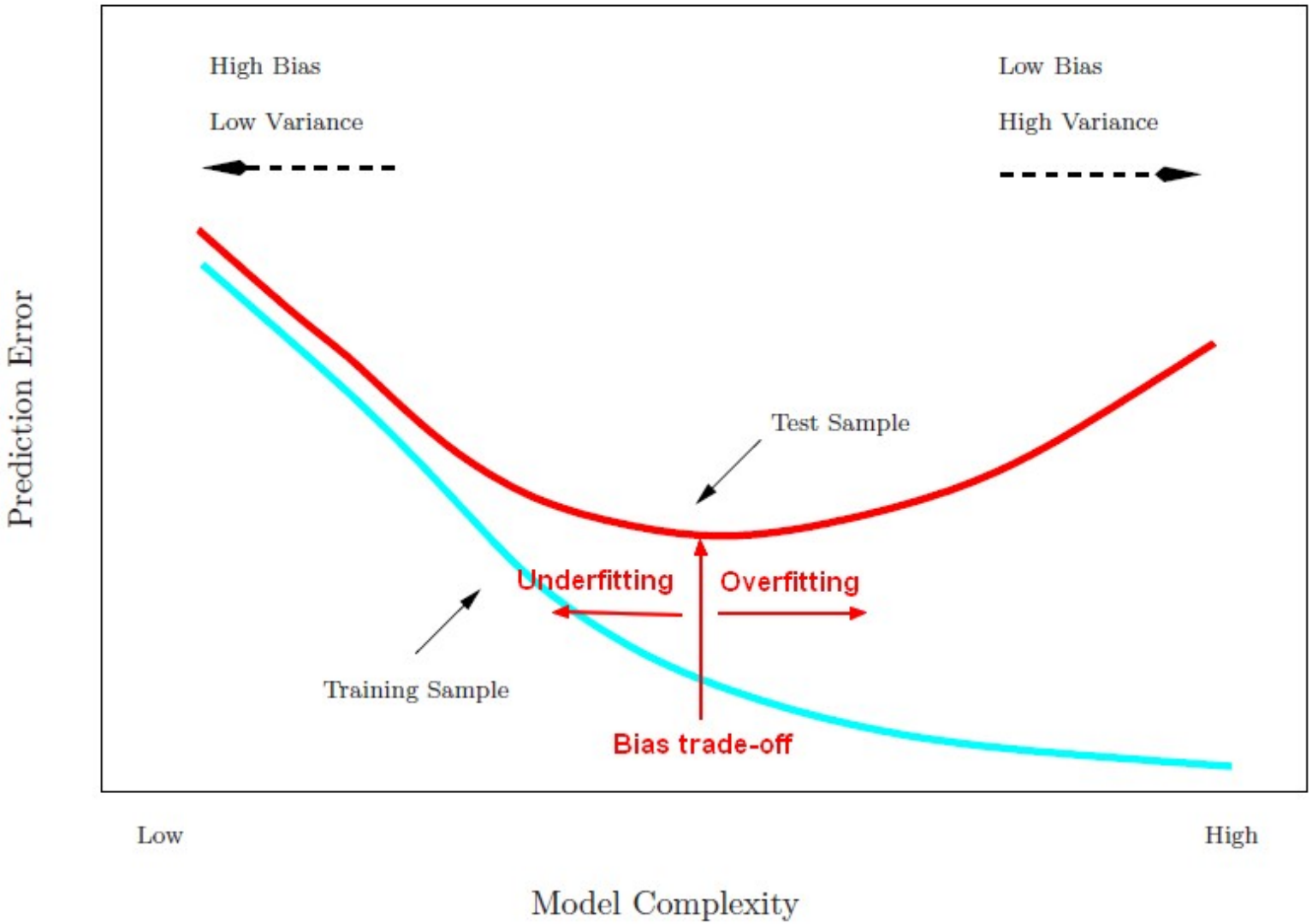


A future challenge: building databases

- GDB-9/QM9 is the most commonly-used training set, the equivalent of the MNIST set of ~10,000 labeled handwritten digits.
- It is now suffering from the same problem as MNIST: it is too simple and not representative of real-world training cases (molecules).
- Analogy: Pople basis sets (6-31G and derivatives) are still extremely common, not even because we don't know better, but because we "need to compare to past work".
- If a *general and transferable* ML model fails on GDB-9, that is a warning sign, but the above cannot be a reason against extending deeper into chemical space for ML model training.

Backup Slides

Definition of overfitting



What is a "one-hot" vector?

> *(...) a group of bits among which the legal combinations of values are only those with a single high (1) bit and all the others low (0).*

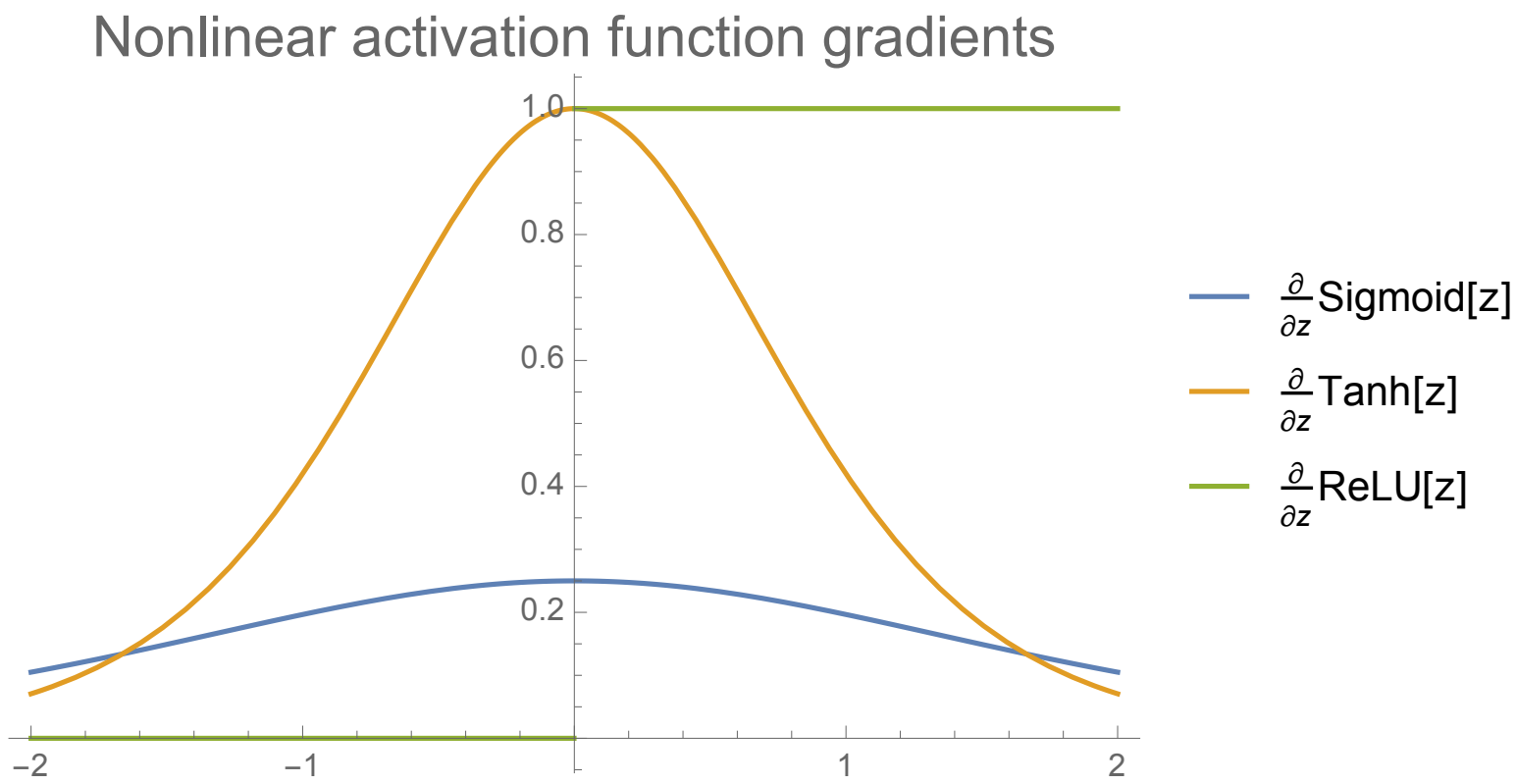
For a variable that can take a finite set of n values, it can be represented as binary vector of length n.

Feature	Description	Size
Atom type	H, C, N, O, or F (one-hot)	5

If atom type is the first feature, this is the third atom in the input, and the element is oxygen, then $x_1^{(3)} = (0, 0, 0, 1, 0)$.

Gradients of nonlinear activation functions

Out[12]=



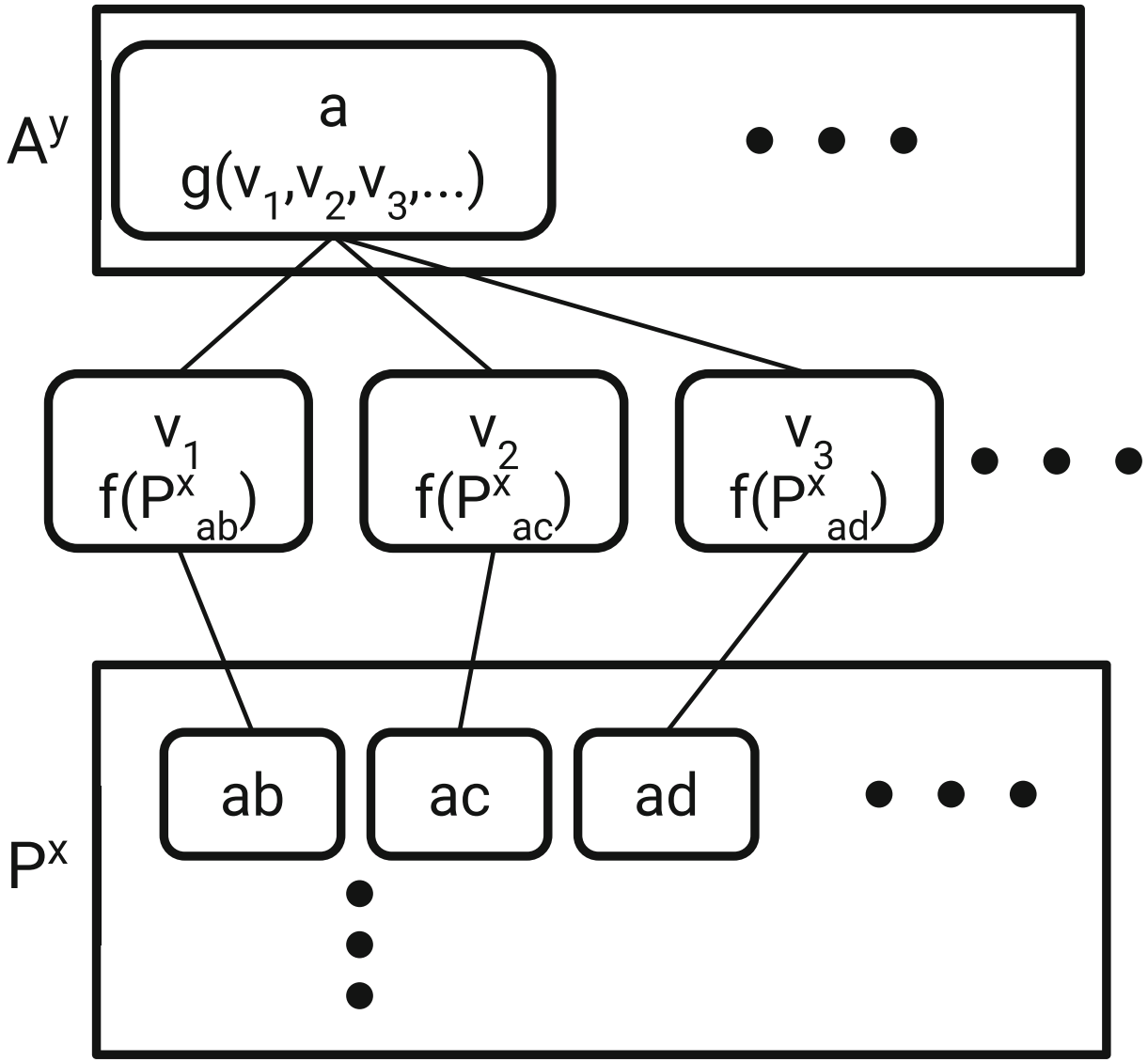
Backpropagation algorithm

for computing weight updates in a fully-connected neural network:

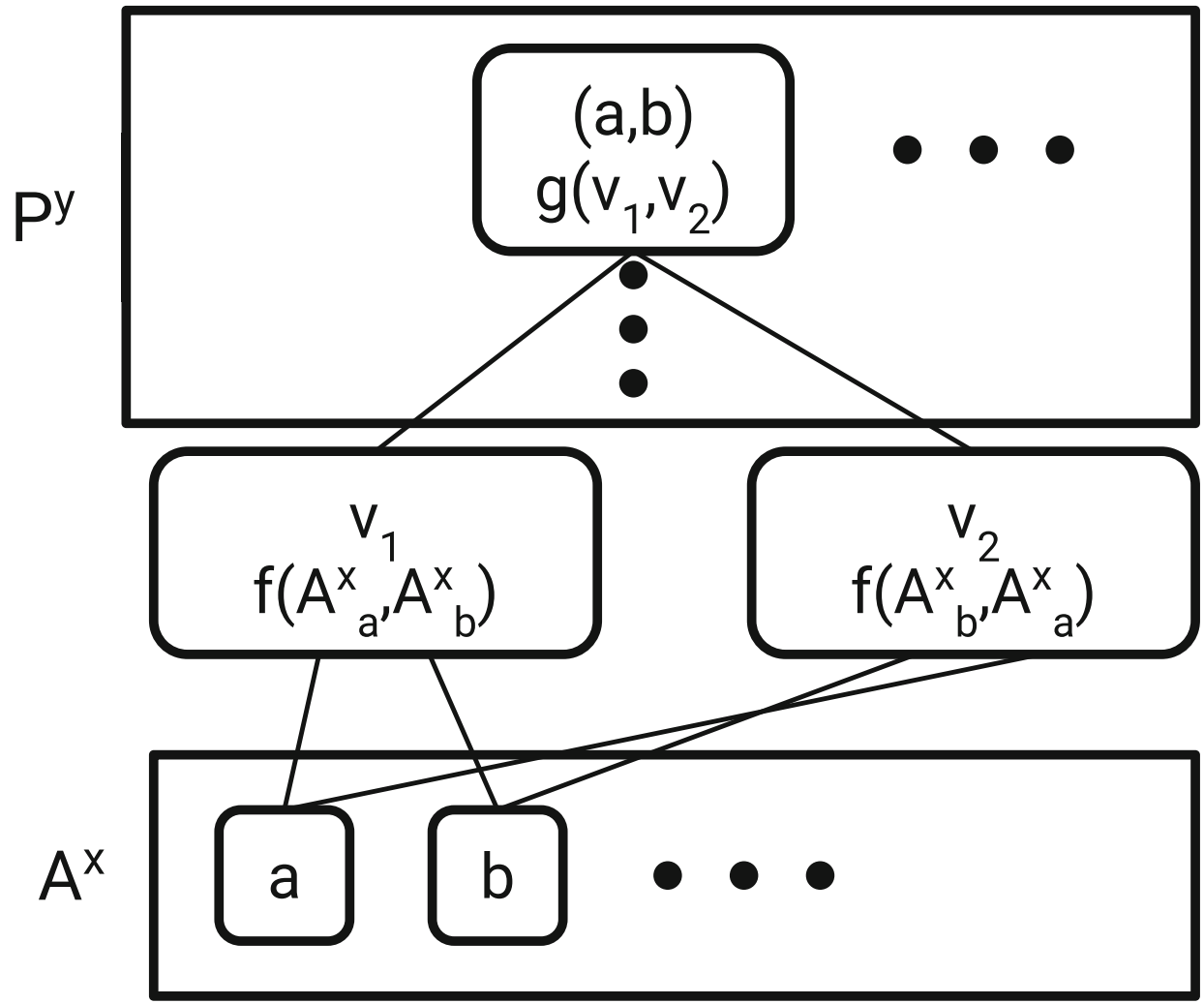
Training set $\leftarrow \{(x^{(1)}, y^{(1)}), ..., (x^{(m)}, y^{(m)})\}$
$\Delta_{ij}^{(l)} \leftarrow 0$ for all l, i, j
for training example $t \leftarrow 1, m$ **do**
    $a^{(1)} \leftarrow x^{(t)}$
    Perform forward propagation to compute $a^{(l)}$ for $l = 2, 3, ..., L$
    $\delta^{(L)} \leftarrow a^{(L)} - y^{(t)}$ ▷ Initialize backpropagation routine at output layer.
    for $l \leftarrow L - 2, 2$ **do** ▷ Work backwards through layers.
        $g'(s^{(l)}) \leftarrow a^{(l)} \odot (1 - a^{(l)})$
        $\delta^{(l)} \leftarrow ((w^{(l)})^T \delta^{(l+1)}) \odot g'(s^{(l)})$
        $\Delta^{(l)} \leftarrow \Delta^{(l)} + \delta^{(l+1)}(a^{(l)})^T$
        $D_{ij}^{(l)} \leftarrow \frac{1}{m}\Delta_{ij}^{(l)} + \lambda w_{ij}^{(l)}$
    end for
end for
$\frac{\partial}{\partial w_{ij}^{(l)}} J(w) \leftarrow D_{ij}^{(l)}$

The regularization term with hyperparameter λ is set to zero if $j = 0$ (the bias node).

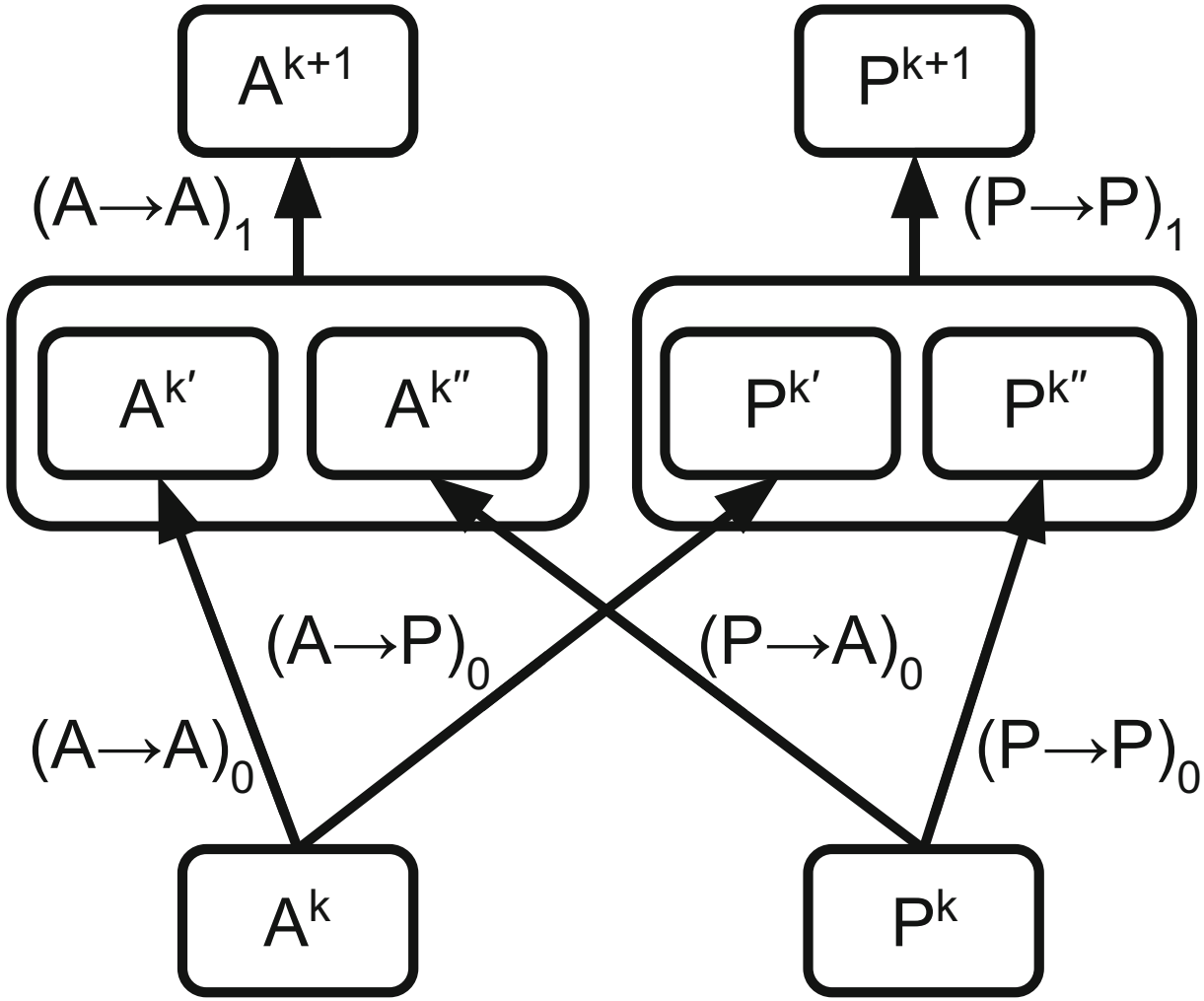
Graph convolution architecture: pairs to atoms



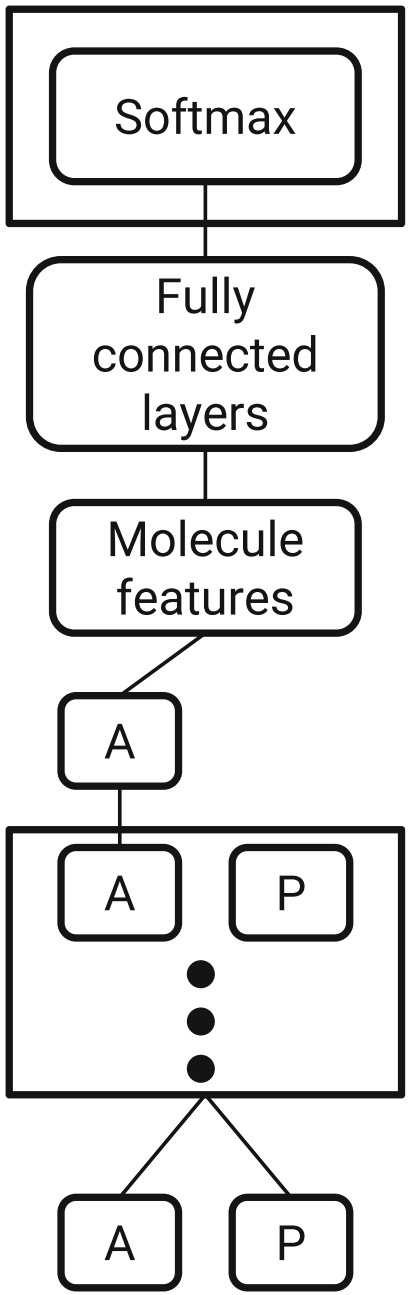
Graph convolution architecture: atoms to pairs



Graph convolution architecture: Weave module



Full graph convolution architecture

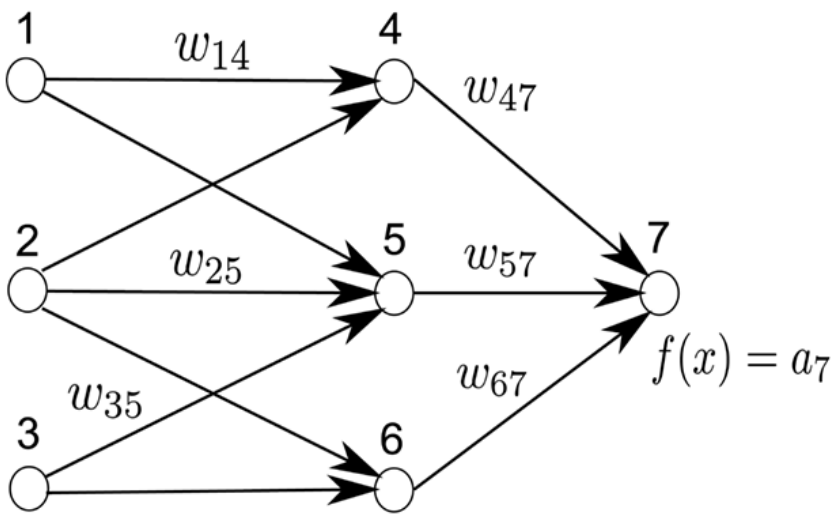


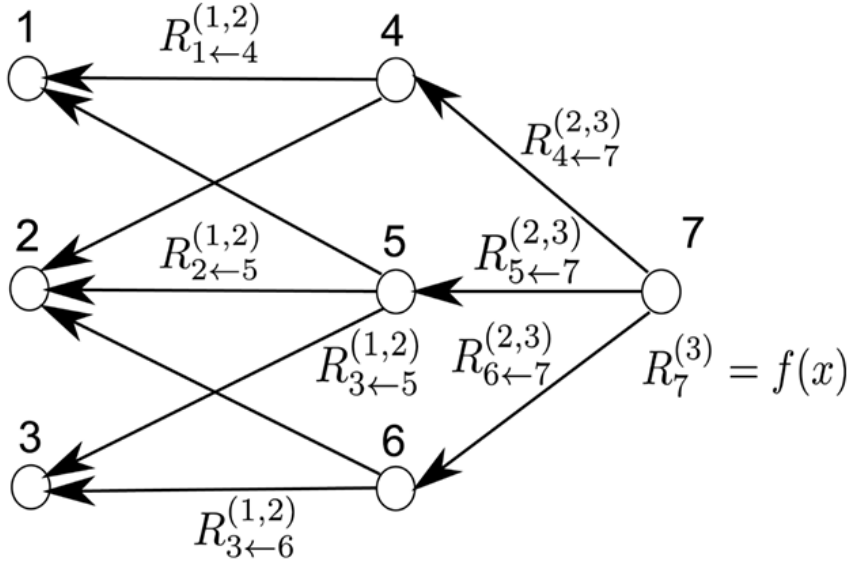
Task-specific

Final convolution

Feature weave modules

Relationship between feed-forward pass and LRP pass





Hyperpolarizability equations from paper

When the dipole moment coincides with the j-axis, we have

$$\beta_{\parallel} = \frac{3}{5}\beta_j = \frac{1}{5}\sum_{i=x,y,z}(\beta_{iij} + \beta_{iji} + \beta_{jii}),$$

or in the general case,

$$\beta_{\parallel} = \frac{3}{5|\mu|}\sum_{i,j=x,y,z}\beta_{iij}\mu_j,$$

where

$$\beta_{ijk} = \langle\langle \mu_i; \mu_j, \mu_k \rangle\rangle .$$

"Coulomb" matrix

$$M_{IJ} = \begin{cases} 0.5 Z_I^{2.4} & \text{for } I = J \\ \frac{Z_I Z_J}{|\mathbf{R}_I - \mathbf{R}_J|} & \text{for } I \neq J. \end{cases}$$

Here, off-diagonal elements correspond to the Coulomb repulsion between atoms I and J, while diagonal elements encode a polynomial fit of atomic energies to nuclear charge.