

Предсказание энергии связи

Рак Алексей

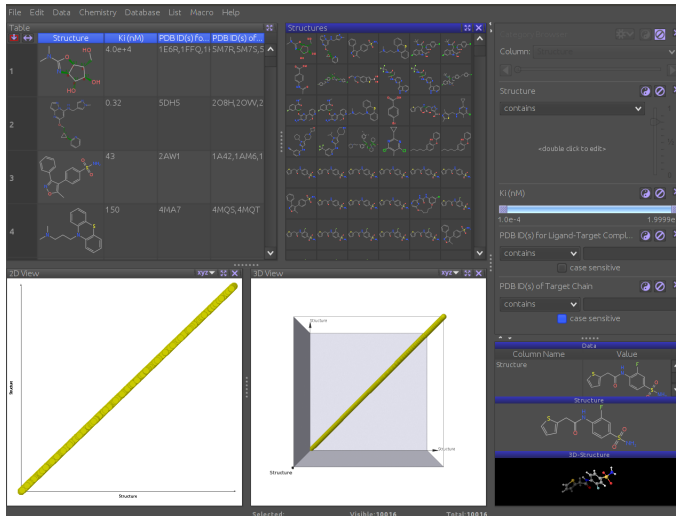
13 декабря 2017 г.

1 Используемые данные

2 Теория

- Нейронные Сети
- Свёрточные нейронные сети
- Атомная свёрточная сеть

Используемые данные



Какие данные используются:

Какие данные используются:

- Структура лиганда

Какие данные используются:

- Структура лиганда
- K_i (энергия связи)

Какие данные используются:

- Структура лиганда
- K_i (энергия связи)
- Id белка в pdb

Какие данные используются:

- Структура лиганда
- K_i (энергия связи)
- Id белка в pdb
- Id комплекса в pdb

Какие данные используются:

- Структура лиганда
- K_i (энергия связи)
- Id белка в pdb
- Id комплекса в pdb

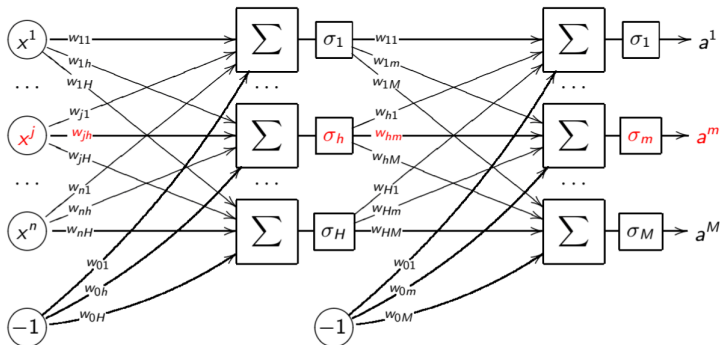
Нейронные сети

Пусть для общности $Y = \mathbb{R}^M$, для простоты слоёв только два.

входной слой,
 n признаков

скрытый слой,
 H нейронов

выходной слой,
 M нейронов



Вектор параметров модели $w \equiv (w_{jh}, w_{hm}) \in \mathbb{R}^{Hn+H+MH+M}$.

- Позволяют решать широкий класс задач

- Позволяют решать широкий класс задач
- Метод обратного распространения ошибки - основанный на простейшем правиле дифференцирования сложной функции, позволяет учить нейронные сети

- Позволяют решать широкий класс задач
- Метод обратного распространения ошибки - основанный на простейшем правиле дифференцирования сложной функции, позволяет учить нейронные сети
- Очень много способов улучшать сходимость и качество нейросети:

- Позволяют решать широкий класс задач
- Метод обратного распространения ошибки - основанный на простейшем правиле дифференцирования сложной функции, позволяет учить нейронные сети
- Очень много способов улучшать сходимость и качество нейросети:
 - Регуляризация

- Позволяют решать широкий класс задач
- Метод обратного распространения ошибки - основанный на простейшем правиле дифференцирования сложной функции, позволяет учить нейронные сети
- Очень много способов улучшать сходимость и качество нейросети:
 - Регуляризация
 - Перетасовка объектов

- Позволяют решать широкий класс задач
- Метод обратного распространения ошибки - основанный на простейшем правиле дифференцирования сложной функции, позволяет учить нейронные сети
- Очень много способов улучшать сходимость и качество нейросети:
 - Регуляризация
 - Перетасовка объектов
 - Разные градиентные методы

- Позволяют решать широкий класс задач
- Метод обратного распространения ошибки - основанный на простейшем правиле дифференцирования сложной функции, позволяет учить нейронные сети
- Очень много способов улучшать сходимость и качество нейросети:
 - Регуляризация
 - Перетасовка объектов
 - Разные градиентные методы
 - Адаптивный learning rate

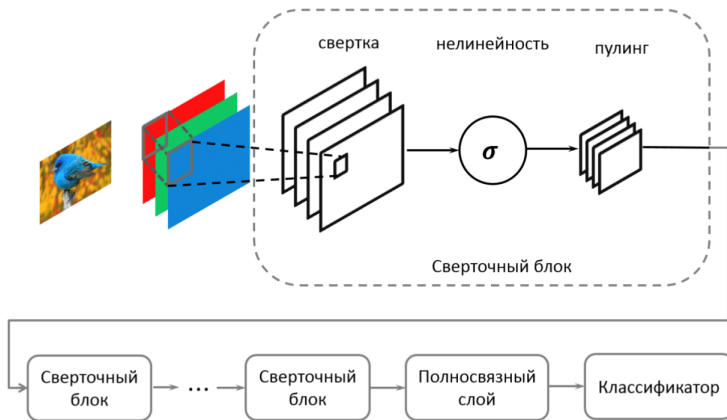
- Позволяют решать широкий класс задач
- Метод обратного распространения ошибки - основанный на простейшем правиле дифференцирования сложной функции, позволяет учить нейронные сети
- Очень много способов улучшать сходимость и качество нейросети:
 - Регуляризация
 - Перетасовка объектов
 - Разные градиентные методы
 - Адаптивный learning rate
 - Dropout

- Позволяют решать широкий класс задач
- Метод обратного распространения ошибки - основанный на простейшем правиле дифференцирования сложной функции, позволяет учить нейронные сети
- Очень много способов улучшать сходимость и качество нейросети:
 - Регуляризация
 - Перетасовка объектов
 - Разные градиентные методы
 - Адаптивный learning rate
 - Dropout
 - Разные функции активации

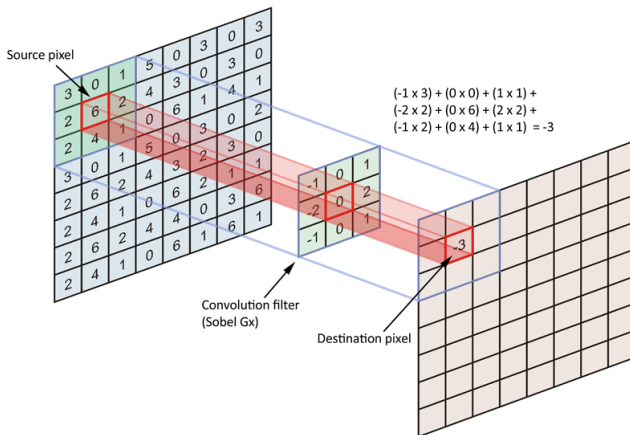
- Позволяют решать широкий класс задач
- Метод обратного распространения ошибки - основанный на простейшем правиле дифференцирования сложной функции, позволяет учить нейронные сети
- Очень много способов улучшать сходимость и качество нейросети:
 - Регуляризация
 - Перетасовка объектов
 - Разные градиентные методы
 - Адаптивный learning rate
 - Dropout
 - Разные функции активации
 - Аугментация (расширение выборки)

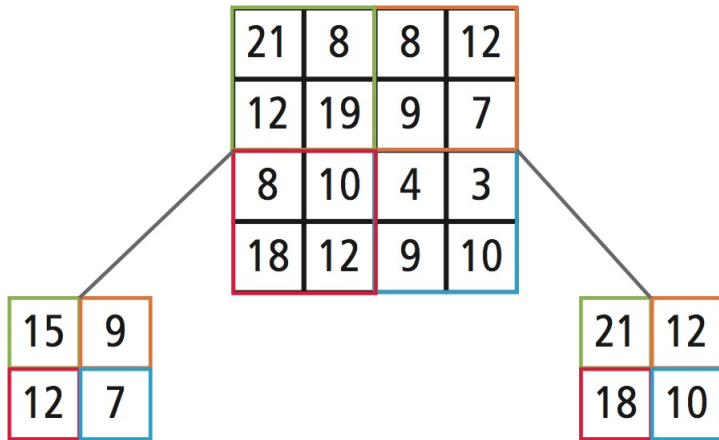
- Позволяют решать широкий класс задач
- Метод обратного распространения ошибки - основанный на простейшем правиле дифференцирования сложной функции, позволяет учить нейронные сети
- Очень много способов улучшать сходимость и качество нейросети:
 - Регуляризация
 - Перетасовка объектов
 - Разные градиентные методы
 - Адаптивный learning rate
 - Dropout
 - Разные функции активации
 - Аугментация (расширение выборки)

Свёрточные нейронные сети



Свёрточный слой

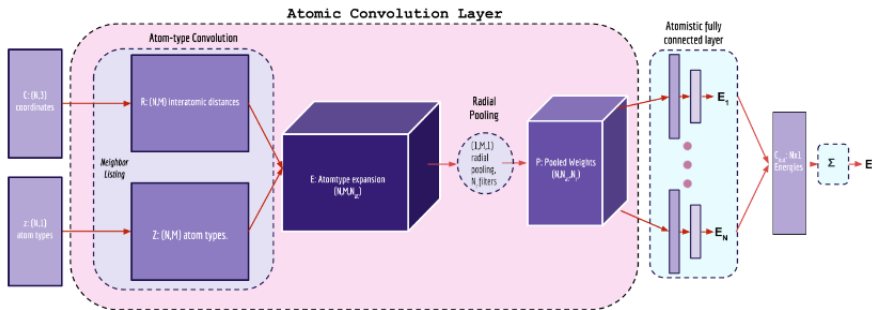




Average Pooling

Max Pooling

Атомная свёрточная сеть



Начальное преобразование данных

- На вход поступает матрица атомов, которая содержит атомный тип каждого атома и его трёхмерные координаты.

Начальное преобразование данных

- На вход поступает матрица атомов, которая содержит атомный тип каждого атома и его трёхмерные координаты.
- Преобразуем эту матрицу в 2 матрицы R и Z размера $(N \times M)$, где N – это число атомов, M – число рассматриваемых соседей (в работе было равно 12).
 - R – матрица расстояний до соседей.
 - Z – матрица атомных типов соседей.

Атомный свёрточный слой

- На входе этого этапа мы получаем 2 матрицы две матрицы R и Z .

Атомный свёрточный слой

- На входе этого этапа мы получаем 2 матрицы две матрицы R и Z .
- N_{at} – число различных атомных типов в данных.

Атомный свёрточный слой

- На входе этого этапа мы получаем 2 матрицы две матрицы R и Z .
- N_{at} – число различных атомных типов в данных.
- $K_{ij}^a = \begin{cases} 1, & Z_{i,j} = N_a \\ 0, & \text{иначе} \end{cases}$

Атомный свёрточный слой

- На входе этого этапа мы получаем 2 матрицы две матрицы R и Z .
- N_{at} – число различных атомных типов в данных.
- $K_{ij}^a = \begin{cases} 1, & Z_{i,j} = N_a \\ 0, & \text{иначе} \end{cases}$
- $(K * R)_{ij}^a = R_{ij} K_{ij}^a$

- На входе этого этапа мы получаем 2 матрицы две матрицы R и Z .
- N_{at} – число различных атомных типов в данных.
- $K_{ij}^a = \begin{cases} 1, & Z_{i,j} = N_a \\ 0, & \text{иначе} \end{cases}$
- $(K * R)_{ij}^a = R_{ij} K_{ij}^a$
- Применив такие операции мы получаем матрицу E размера (N, M, N_{at})

Атомный пулинговый слой

- На входе этого этапа мы получаем матрицу E размера (N, M, N_{at}) .

- На входе этого этапа мы получаем матрицу E размера (N, M, N_{at}) .
- $f_s(r_{ij}) = \exp\left(-\frac{(r_{ij}-r_s)^2}{\sigma_s^2} f_c(r_{ij})\right)$

- На входе этого этапа мы получаем матрицу E размера (N, M, N_{at}) .
- $f_s(r_{ij}) = \exp\left(-\frac{(r_{ij}-r_s)^2}{\sigma_s^2} f_c(r_{ij})\right)$
- $f_c(r_{ij}) = \begin{cases} \frac{1}{2} \cos\left(\frac{\pi r_{ij}}{R_c}\right), & 0 < r_{ij} < R_c \\ 0, & r_{ij} \geq R_c \end{cases}$

- На входе этого этапа мы получаем матрицу E размера (N, M, N_{at}) .
- $f_s(r_{ij}) = \exp\left(-\frac{(r_{ij}-r_s)^2}{\sigma_s^2}\right) f_c(r_{ij})$
- $f_c(r_{ij}) = \begin{cases} \frac{1}{2} \cos\left(\frac{\pi r_{ij}}{R_c}\right), & 0 < r_{ij} < R_c \\ 0, & r_{ij} \geq R_c \end{cases}$
- $P_{i,n_a,n_r} = \beta_n \sum_{i=1}^M f_{n_r}(E_{ijn_a}) + b_{n_r}$

- На входе этого этапа мы получаем матрицу E размера (N, M, N_{at}) .
- $f_s(r_{ij}) = \exp\left(-\frac{(r_{ij}-r_s)^2}{\sigma_s^2} f_c(r_{ij})\right)$
- $f_c(r_{ij}) = \begin{cases} \frac{1}{2} \cos\left(\frac{\pi r_{ij}}{R_c}\right), & 0 < r_{ij} < R_c \\ 0, & r_{ij} \geq R_c \end{cases}$
- $P_{i,n_a,n_r} = \beta_n \sum_{i=1}^M f_{n_r}(E_{ijn_a}) + b_{n_r}$
- В итоге мы получаем матрицу P размера (N, N_a, N_r)

- На входе этого этапа мы получаем матрицу E размера (N, M, N_{at}) .
- $f_s(r_{ij}) = \exp\left(-\frac{(r_{ij}-r_s)^2}{\sigma_s^2} f_c(r_{ij})\right)$
- $f_c(r_{ij}) = \begin{cases} \frac{1}{2} \cos\left(\frac{\pi r_{ij}}{R_c}\right), & 0 < r_{ij} < R_c \\ 0, & r_{ij} \geq R_c \end{cases}$
- $P_{i,n_a,n_r} = \beta_n \sum_{i=1}^M f_{n_r}(E_{ijn_a}) + b_{n_r}$
- В итоге мы получаем матрицу P размера (N, N_a, N_r)
- r_s, σ_s – обучаемы параметры.

- На входе этого этапа мы получаем матрицу E размера (N, M, N_{at}) .
- $f_s(r_{ij}) = \exp\left(-\frac{(r_{ij}-r_s)^2}{\sigma_s^2}\right) f_c(r_{ij})$
- $f_c(r_{ij}) = \begin{cases} \frac{1}{2} \cos\left(\frac{\pi r_{ij}}{R_c}\right), & 0 < r_{ij} < R_c \\ 0, & r_{ij} \geq R_c \end{cases}$
- $P_{i,n_a,n_r} = \beta_n \sum_{i=1}^M f_{n_r}(E_{ijn_a}) + b_{n_r}$
- В итоге мы получаем матрицу P размера (N, N_a, N_r)
- r_s, σ_s – обучаемы параметры.
- β, b – постоянные выбираемые до обучения.

- На входе этого этапа мы получаем матрицу E размера (N, M, N_{at}) .
- $f_s(r_{ij}) = \exp\left(-\frac{(r_{ij}-r_s)^2}{\sigma_s^2}\right) f_c(r_{ij})$
- $f_c(r_{ij}) = \begin{cases} \frac{1}{2} \cos\left(\frac{\pi r_{ij}}{R_c}\right), & 0 < r_{ij} < R_c \\ 0, & r_{ij} \geq R_c \end{cases}$
- $P_{i,n_a,n_r} = \beta_n \sum_{i=1}^M f_{n_r}(E_{ijn_a}) + b_{n_r}$
- В итоге мы получаем матрицу P размера (N, N_a, N_r)
- r_s, σ_s – обучаемы параметры.
- β, b – постоянные выбираемые до обучения.

Атомный полносвязный слой

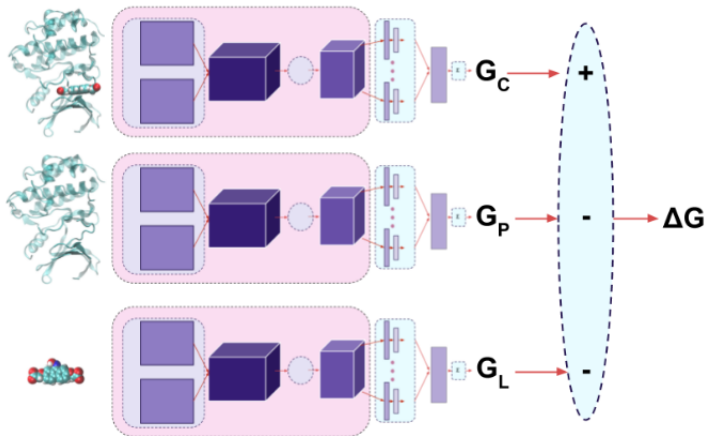
- На входе этого этапа мы получаем матрицу E размера (N, N_{at}, N_r) .

- На входе этого этапа мы получаем матрицу E размера (N, N_{at}, N_r) .
- N_r – это число различных пулинговых фильтров.

- На входе этого этапа мы получаем матрицу E размера (N, N_{at}, N_r) .
- N_r – это число различных пулинговых фильтров.
- Затем строится двухслойная нейронная сеть, одинаковая для всех атомов, и прогоняют через неё матрицу каждого атома $(1, N_{at}, N_r)$

- На входе этого этапа мы получаем матрицу E размера (N, N_{at}, N_r) .
- N_r – это число различных пулинговых фильтров.
- Затем строится двухслойная нейронная сеть, одинаковая для всех атомов, и прогоняют через неё матрицу каждого атома $(1, N_{at}, N_r)$
- Для каждого атома получаем по скаляру, которые суммируем, это и есть ответ.

Энергия связи



Коэффициент детерминации

$$R^2 = 1 - \frac{\hat{\sigma}^2}{\hat{\sigma}_y^2} = 1 - \frac{SS_{res}/n}{SS_{tot}/n} = 1 - \frac{SS_{res}}{SS_{tot}}$$

$$SS_{res} = \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

$$SS_{tot} = \sum_{i=1}^n (y_i - \bar{y})^2$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

y_i – фактическое значение, \hat{y}_i – расчётное значение

