3Q 研究プロジェクト（石田研）

提出日：2018 年 10 月 22 日

系／学科／類：情報工学系

学籍番号：16B13354

氏名：星野　シンジ

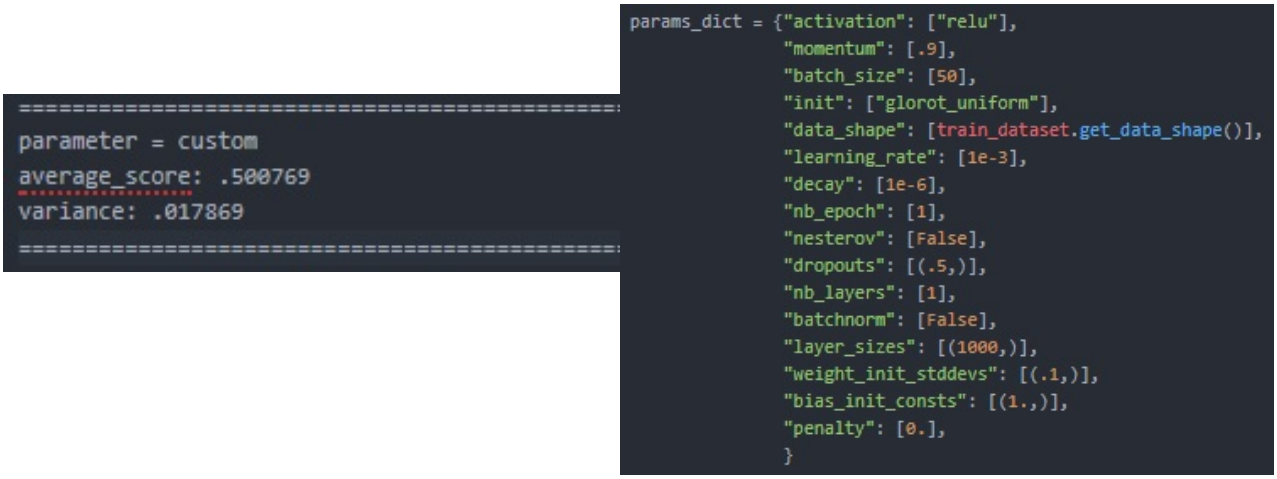


図 1: デフォルトパラメーターでの実行結果

1　機械学習を用いた薬剤活性の予測

python 向けのライブラリである deepchem をインストールし、そのチュートリアルのひとつである「Multitask Networks on MUV」をベースにして、薬剤活性の予測を行った。

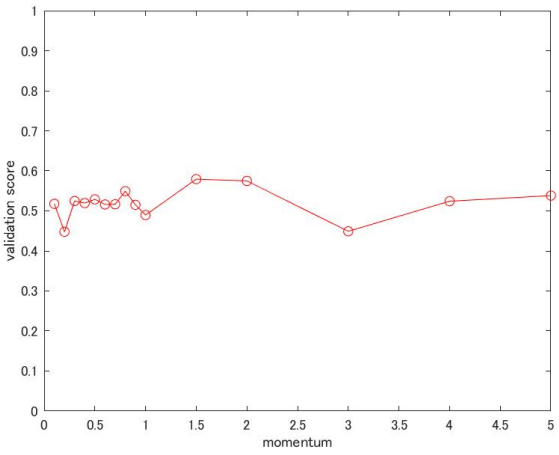
まず、チュートリアルに入っているハイパーパラメータで５０回実行したときの結果は図 1 のようになった。

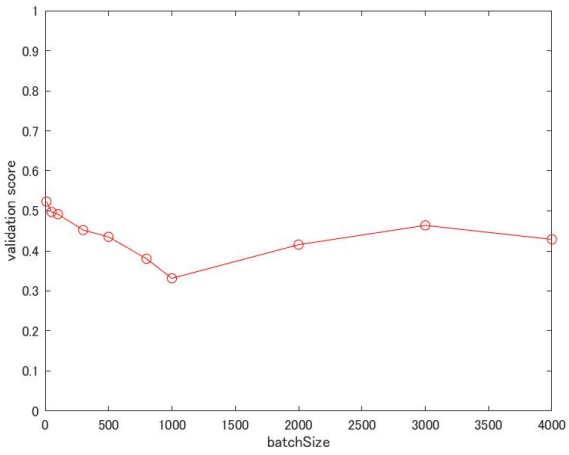
このスコアが更に良くなるように、ハイパーパラメータを調整したい。そこで、もとのハイパーパラメータの変数を一つずつ変更し、十回実行したときの平均スコアがどのように変化するかを調べた。一回実行しただけだと、偶然いい値が出る場合があるので、十回実行したスコアの平均と分散を計算した。平均の値は、図 2 のようになった。

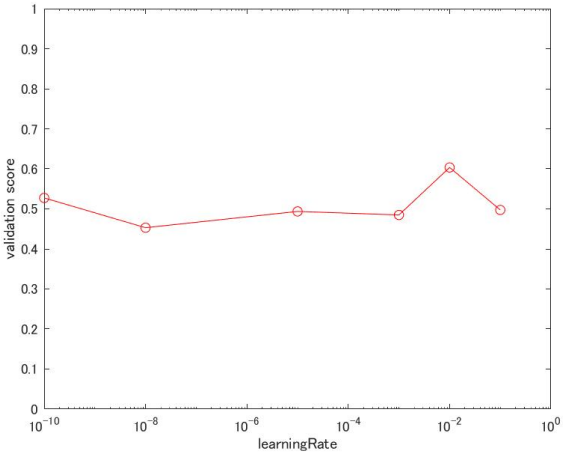
ここまで、実験を行ったところで、より適切なハイパーパラメーターを設定できるように、それぞれのパラメーターの意味について調べて表 1 にまとめた。

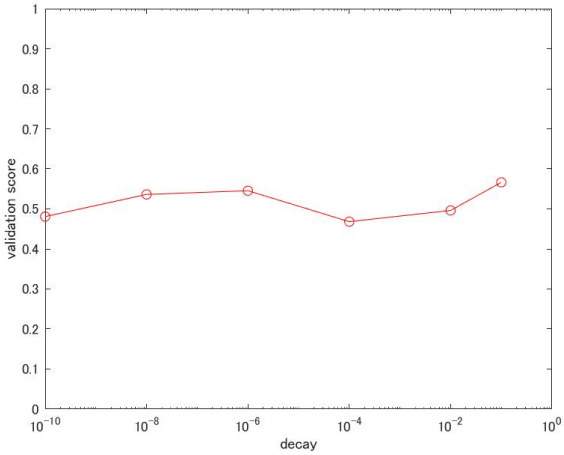
これらの情報に基づき、一番良いパラメーターの値を分散も考慮しながら様々な組み合わせで試したところ、図 3 のようなパラメーターでスコアの平均をを 0.613569 まで上げることができた。しかも、十回の思考での分散が 0.005626 ともともとのパラメーターのときよりも小さくなっているので、安定して 6 割程度の制度を得た。この時、とくに影響のあったパラメーターは、nb_epoch と nb_layers であった。epoch は大きめに設定してトレーニングを繰り返すことを行ったために制度が上がったと考える。また、layers を適切な値にすることによって 1 レイヤーの時より複雑なモデルになったが、過学習を起こすほどではないようになったと考える。それぞれのパラメーターのスコアへの影響を考えると、これよりも精度を上げるためには、レイヤーの数とそれぞれのレイヤーでのノードの数をさらに最適化することに注目したほうがいいと考えた。

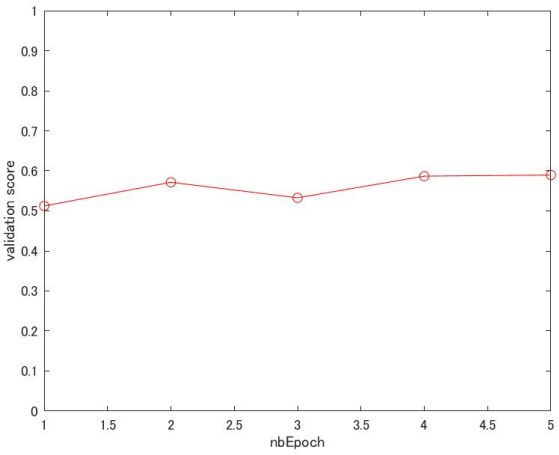
層の数を増やした状態で、さらに層ごとに変数を変化させたらどうなるかを実験したかったが、計算に時間がかかりすぎるので行わなかった。しかし、手動でいくつか試してみたところさらに良

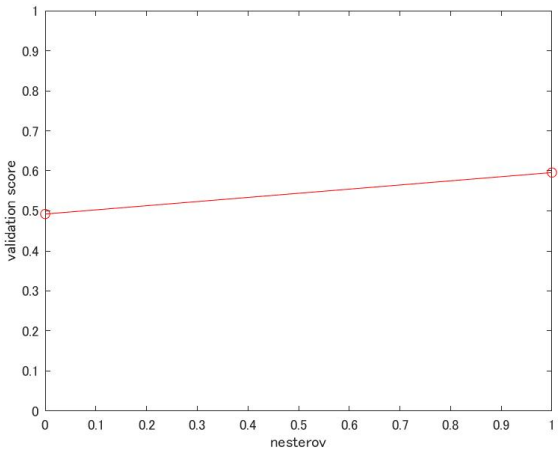

validation score
momentum

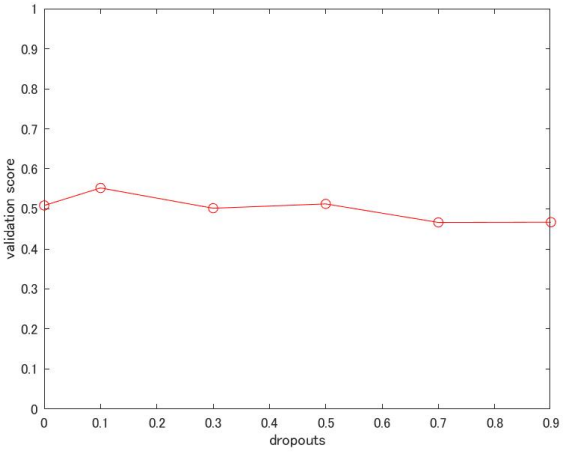

validation score
batchSize

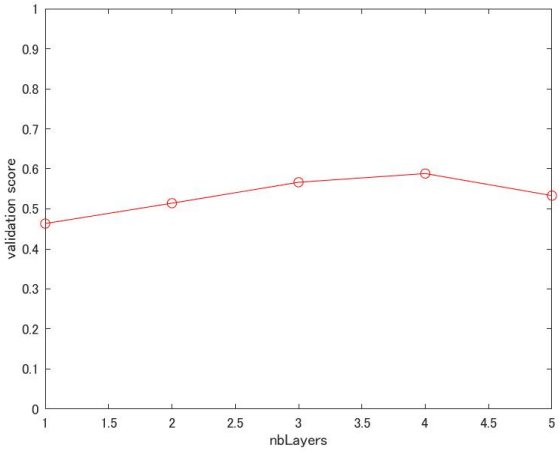

validation score
learningRate

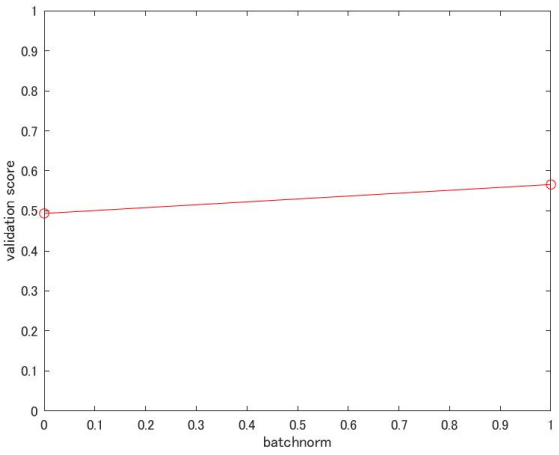

validation score
decay

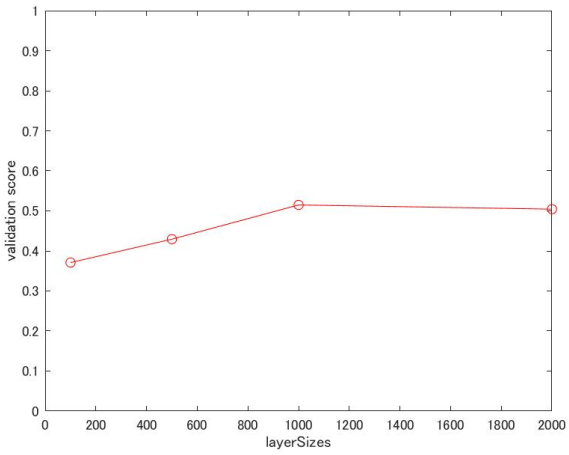

validation score
nbEpoch
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1
1.5
2
2.5
3
3.5
4
4.5
5

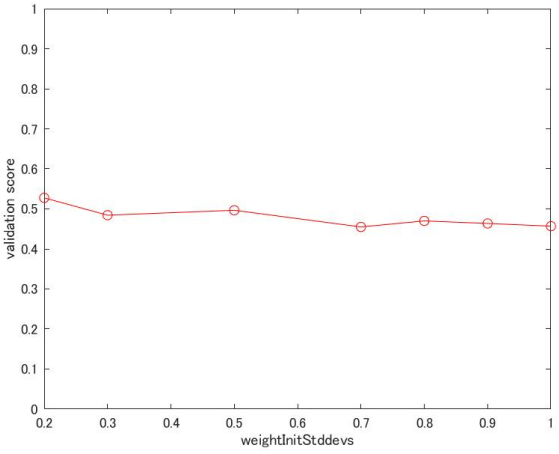

validation score
nesterov
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

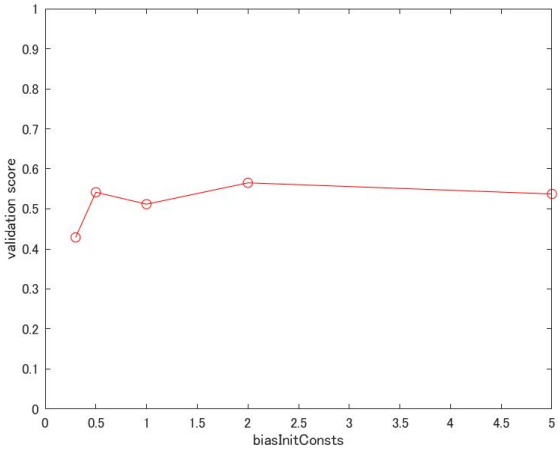

validation score
dropouts
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


validation score
nbLayers
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1
1.5
2
2.5
3
3.5
4
4.5
5


validation score
batchnorm


validation score
layerSizes


validation score
weightInitStddevs


validation score
biasInitConsts

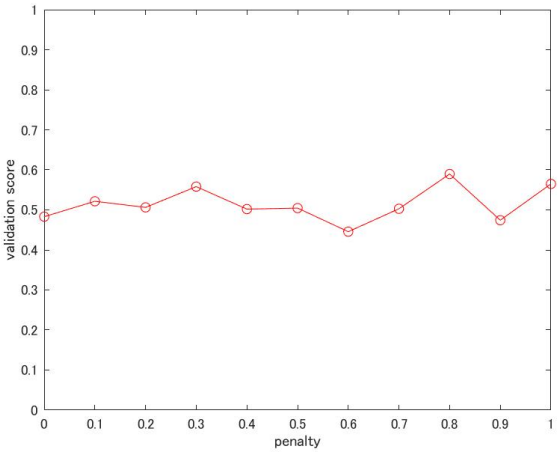


図 2: 実験結果

```
==========================================
parameter = custom
average_score: .613569
variance: .005626
==========================================
```

```
params_dict = {"activation": ["relu"],
               "momentum": [0.8],
               "batch_size": [50],
               "init": ["glorot_uniform"],
               "data_shape": [train_dataset.get_data_shape()],
               "learning_rate": [1e-2],
               "decay": [1e-6],
               "nb_epoch": [10],
               "nesterov": [True],
               "dropouts": [(.1, .1, .1, .1, .1)],
               "nb_layers": [5],
               "batchnorm": [True],
               "layer_sizes": [(1000, 1000, 1000, 1000, 1000)],
               "weight_init_stddevs": [(.2, .2, .2, .2, .2,)],
               "bias_init_consts": [(2., 2., 2., 2., 2.,)],
               "penalty": [0.],
              }
```

図 3: パラメーターとその実行結果

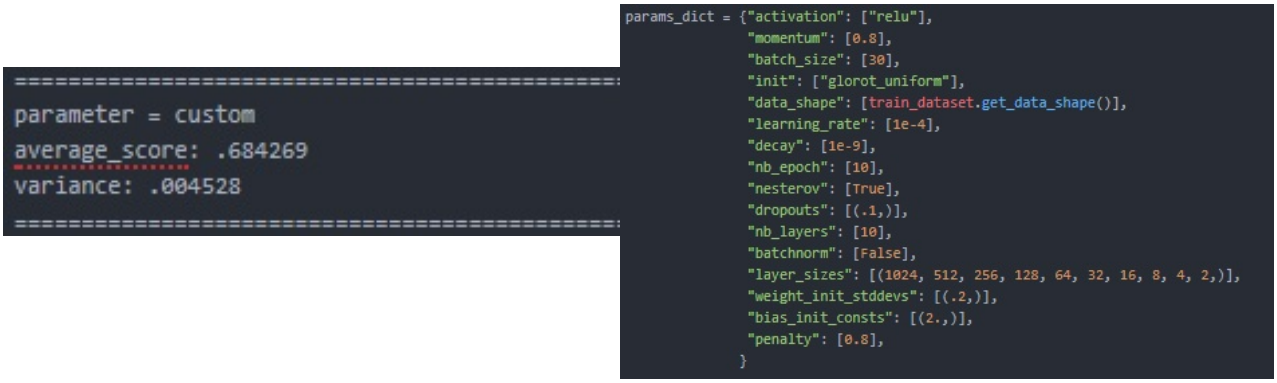


図 4: 一番良かったパラメーターとその実行結果

いパラメーターの組み合わせを見つけることができたので、図 4 にパラメーターとその実行結果を示しておく。層とノードの設定はそのままにし、ほかのパラメーターを変化させてみたが、３%程度しか精度が上がらずかつ分散が大きくなり結果が不安定になった。したがって、ここまで精度が上がったのは、ニューラルネットワークの層とノードの数がちょうどいいものになったからだと考えている。

結果として、ハイパーパラメーターの調整を行った結果、スコアの平均を 0.684269（分散は 0.004528) まで上げることに成功した。

パラメーター名	説明
momentum	エラー関数の大域的最小値を探そうとする際に、局所解を求めるようになっている。この値が大きければ大きいほど、アルゴリズムの「勢い」のようなものが大きくなり、ある局所解が求まったとしても、より最適な解を探そうとするようになる。[5]
batch_size	学習アルゴリズムが内部モデルのパラメーターを調整するまでに取り込むデータの数を示す。つまり、この数だけデータをとるごとにエラーチェックをし、修正を行う。[4]
learning_rate, decay	アルゴリズムがデータから学習する際に、新しいデータの取り込みに対する感度を示す。learning_rate の値が大きければ大きいほど、新しいデータに強く適応するようになる。また、データを取り込むにつれてあとから入ってくるデータの重みが小さくなっていくが、decay はその小さくなっていく速さを示す。[6]
nb_epoch	学習アルゴリズムがトレーニングデータのセットを何度学ぶかを示すパラメーター。[4]
dropout	過激適合を避けるために、ノード間のウェイトを均等にする必要がある。そのために、ランダムにレイヤーのノードをオフにする、dropout というものが行われるが、これはそれが起きる頻度を示す。[7]
nb_layers, layer_sizes	ニューラルネットワークの層の数、および各層のノードの数を示す。基本的には、大きいほど性能が上がる。 [9]
weight_init_stddevs	重みの初期値を示す。これは、訓練中に調整されていく。[8]
bias_init_consts	バイアスの初期値を示す。ウェイトと同じく、これも訓練中に調整されていく。[8]
penalty	正則化を行う際の、極端なデータの重みに対するペナルティーを示す。これが大きいと、極端なデータの重みがより小さくなり、モデルに影響を与えにくくなる。[3]

表 1: パラメーターの説明

2　MUV データセットについて

まず、このデータ・セットがなんのためにあるのかについて、論文 [1] を読んでわかったことを以下で説明する。

現代の薬学の世界では、バーチャールスクリーニングを行うことにより、薬学的に役に立つであろう物質を選び出すことが行われている。ところが、最近の研究により分子構造に基づいたバーチャルスクリーニング（SBVS）は、バリデーションで用いられるデータセットが結果に大きな影響を及ぼすことがわかった。特に、データ・セット間で分子量・水素結合の数等の「シンプル」な性質が大きく異なっていると、スコアが上がったかのように見えることがわかっている。したがって、ある一定の範囲に収まる「シンプル」な性質を持つものを集めたデータ・セットでなければ、正しいバリデーションとは言えず、バイアスがかかってしまうと考えられる。今回用意されているデータ・セットは、そのようなバイアスがかからないように、集められたものであり、これは Maximum Unbiased Validation(最もバイアスが小さいバリデーション) メソッドを用いたバーチャルスクリーニングを可能にする。

次に、データ・セットに含まれていたタンパク質のうちの一つに注目し、それに対して活性のある化合物と活性のない化合物を挙げる。これらのデータは、PubChem[2] より入手したものである。

今回注目したタンパク質は、MVU のデータ・セットに含まれる、MUV-600 というものである。MUV-600 の構造は図 5 のとおりになっている。

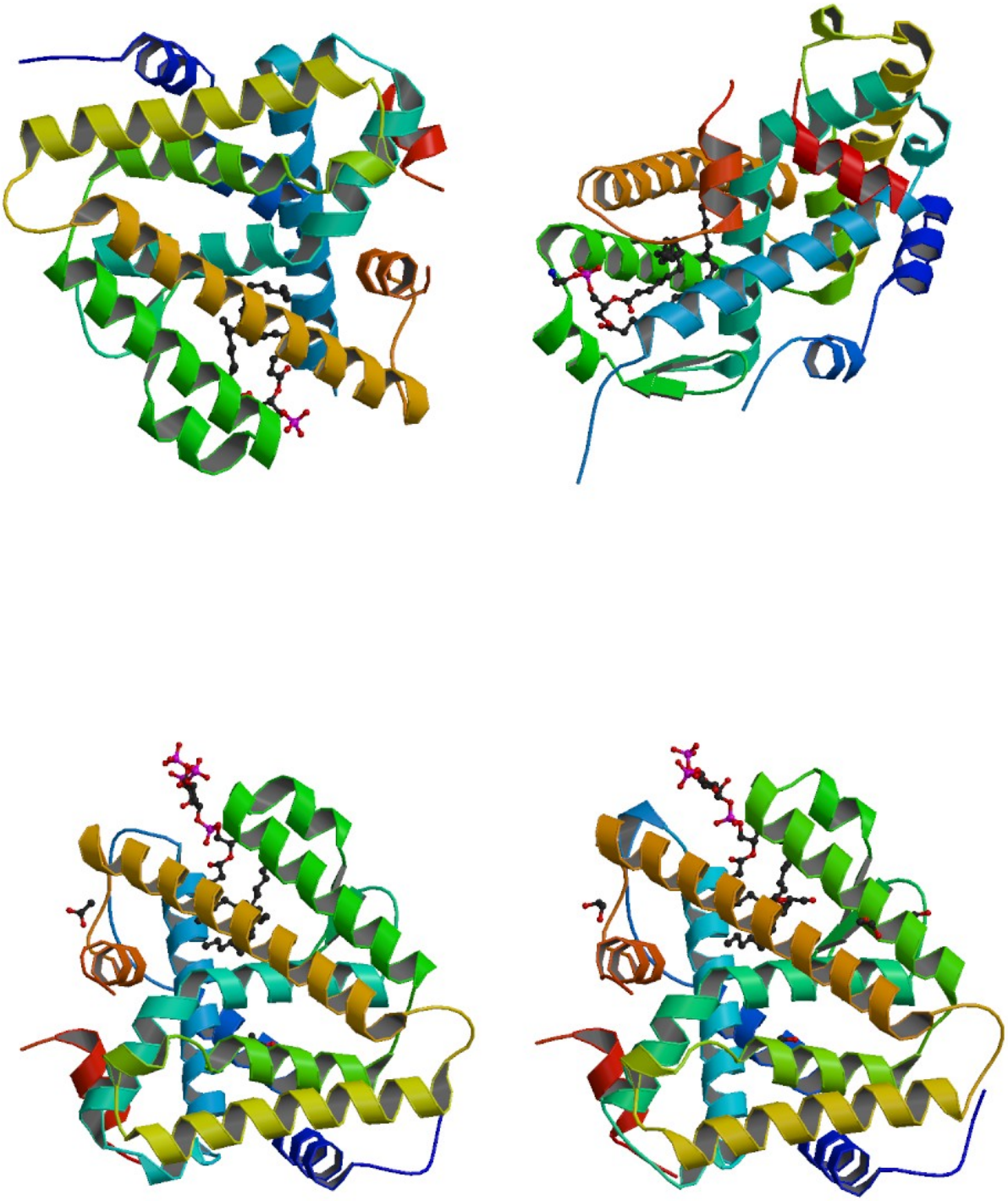


図 5: MUV-600 の構造

このタンパク質に対し、活性のある化合物として図 6 のような構造をしているものが例として挙げられる。

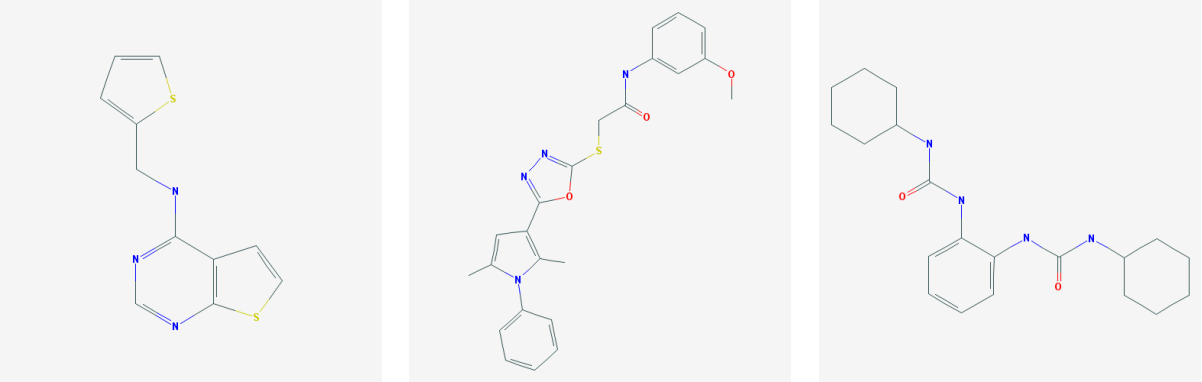


図 6: 活性のある化合物

一方、活性のない化合物としては図 7 のようなものが挙げられる。

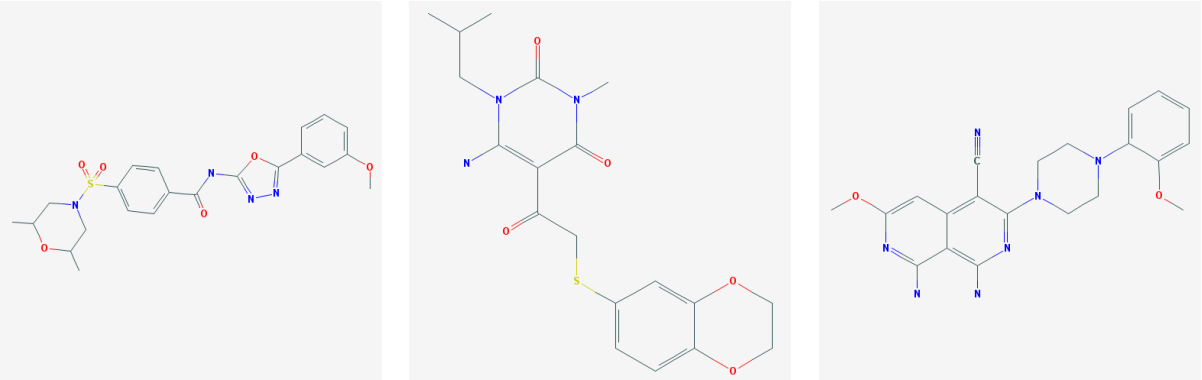


図 7: 活性のない化合物

参考文献

[1] Sebastian G. Rohrer and Knut Baumann, ”Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on PubChem Bioactivity Data, ” https://pubs.acs.org/doi/10.1021/ci8002649 as of Oct. 21 2018.

[2] PubChem MUV-600, https://pubchem.ncbi.nlm.nih.gov/bioassay/600#section=Protocol Protein Target: https://pubchem.ncbi.nlm.nih.gov/target/protein/NP_004950 as of Oct. 21 2018

[3] 過学習を防ぐ方法, https://products.sint.co.jp/aisia/blog/vol1-8 as of Oct. 21 2018

[4] Difference between a batch and an epoch, https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/ as of Oct. 22 2018

[5] Quora: What does momentum mean in neural network, https://www.quora.com/What-does-momentum-mean-in-neural-networks as of Oct. 22 2018

[6] Quora: What is learning rate in neural networks, https://www.quora.com/What-is-the-learning-rate-in-neural-networks as of Oct. 22 2018

[7] Quora: Why exactly does dropout in deep learning work, https://www.quora.com/Why-exactly-does-dropout-in-deep-learning-work as of Oct. 22 2018

[8] StackExchange: What is weight and bias in deep learning, https://datascience.stackexchange.com/questions/19099/what-is-weight-and-bias-in-deep-learning as of Oct. 22 2018

[9] How to configure the number of layers and nodes in a neural network, https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/ as of Oct. 22 2018