IN SILICO DESIGN OF COMPETITIVE INHIBITORS USING DEEP REINFORCEMENT LEARNING

Lukas Dubsik , Gymnazium Brno-Reckovice university, Czech Republic, Contact: xdubs@gyrec.cz,

ARTICLE INFO

Gold medalist in BUCA IMSEF 2022

Consultant: Mgr. Ondrej Schindler – Masaryk

Awarded by Ariaian Young Innovative

Minds Institute , AYIMI

http://www.ayimi.org ,info@ayimi.org

ABSTRACT

The aim of this work is to design a program capable of rapid and accurate design of new drugs. Enzymes and their interacting competitive inhibitors have been chosen as the focus. However, the final model can also be applied to other reactants. Several additional models were also built to achieve the goal, namely NDock which is responsible for faster molecular docking. Several libraries were also created in Python, C# and C++ programming languages to facilitate the manipulation of the files describing each structure. The molecule that is given the highest score using molecular docking (NDock used here) moves on and is edited in a stepwise fashion.

Keywords : *Accurate Design, Molecular Docking, Enzymes ,Competitive Inhibitors*

1. Introduction

Enzymes are proteins that are involved in most biochemical reactions in organisms. The aim of this work is to design a program capable of rapid and accurate design of new drugs based on enzyme inhibitors. An inhibitor is a molecule that binds to an enzyme more strongly than its substrate and therefore prevents the enzymatic reaction of the substrate. Examples of enzymes include the well-known Ibuprofen.

The input of the prediction program is a protein - enzyme and its already known ligand - substrate. Using deep feedback learning methods, the program attempts to find a molecule that binds to the protein more strongly than the provided substrate and thus acts as an inhibitor of the protein. The process is iterative, and at each iteration the inhibitor candidates are suitably adjusted to maximize the binding between the enzyme and the inhibitor.

The program is designed to be general so that it can be easily extended to other enzymatic reaction types. Several additional programs have been designed to achieve the goal. Examples include NDock (Neural Molecular Docking), which is responsible for accelerated molecular docking, or libraries for easier manipulation and handling of molecular structure information files.

The predicted molecules are stable and bear similarity to the experimental inhibitors, but neither their synthesis nor subsequent experimentation was part of the work. The accuracy of the program is close to the accuracy of the classical Autodock 4 method. The main advantage of the developed program is its speed and low computational power cost. The library itself (under the name BIP - Bioinformatic Python) and its various functions, such as force fields, searching for non-covalent interactions or working with chemical files, is applicable in other programs.

2. Method

Today's medicine is usually concerned with synthesising drugs that can be used for the largest possible proportion of the population. However, some fields of medicine require highly specific treatments tailored to the patient. An example is cancer, where the increase in mutations is so rapid and their properties so different that an effective cure for all is almost unattainable. However, a large, highly accurate and sufficiently fast bioinformatics apparatus is needed to design specific drugs.

After the disease-causing protein is discovered and its structure is also found, it can be inserted into the program itself. Here, important information such as electrical charge, aromaticity or free energy is first extracted and then entered into the database. The molecule then goes into a structure modification program that uses deep feedback learning. Here, the structure is iteratively refined over time. As a result, there are several possible, competitive inhibitors. These candidate inhibitors undergo testing to see if they are indeed inhibitors, so as not to further propagate possible errors. Candidates that pass the test are given a score, which is the sum of the free energies obtained using the NDock program. This score tells us how good a competitive inhibitor the inhibitor is. Only the molecule with the highest score is re- entered into the database for structure modification. This iteration continues until the number of iterations specified by the user is completed or the final molecule is found.

The steps of the program, and the results, are summarized below.

2.1. Program to Modify Molecule

At each step of the model, the molecule goes through 3 main processes. First the molecule is treated, then it is classified, and finally it is given a score that tells how good an inhibitor it is. There are 4 separate deep feedback networks operating within the modification, each responsible for one part of the modification (Fig. 1).

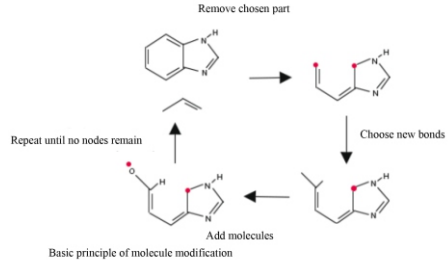


Fig.1: Basic principle of molecule modification

First, all the interconnected parts of the molecule are selected, which, even after their removal, leave the molecule as a single unit. Their maximum size is determined by the user, with a size of 3 for smaller molecules and up to 7 for larger ones. However, the time and computational power requirements increase exponentially with larger size (Fig. 2).

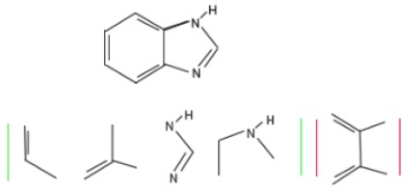


Fig. 2: Allowed and disallowed parts of a molecule, disallowed parts are those that split the molecule into multiple parts.

For each selected part, a decision is then made whether or not to remove it. This is done by the first model. The atoms that have bonded with the removed atoms are then taken as the nodes from which the new part of the molecule will be built. These nodes are then given a score based on the number of bonds they can form (Fig. 3).

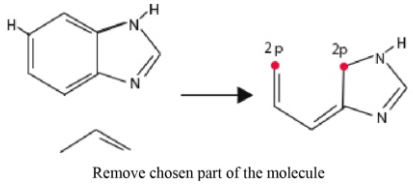


Fig. 3: Remove part and add nodes

Each node is then assigned links that are selected based on both its score and the output of the other model. Atoms or links are then selected to the bonds with the help of the third model. In the case of a link, this node is connected to another node with the help of the fourth model. If any of the newly added atoms has a score greater than zero, that atom then becomes a new node.

The ligand is modified by the model until no nodes remain. This happens gradually by selecting hydrogens as atoms, which always have a score of 0, or by joining nodes. In the initial stages of training, however, the ligand size may grow exponentially, but this can be avoided by setting limits on the start and removing them gradually (Fig. 4).

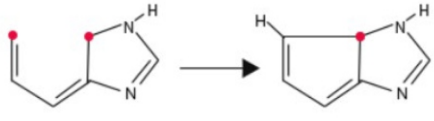


Fig. 4: Possible disappearance of the nodes of the molecule by adding hydrogens or joining, formation also ends when the possible limit of atoms is reached

3. Classification

Using free energy, it is possible to determine how strong the ligand binding is in the active site, but it is not possible to tell if it is an inhibitor or a substrate. If we assume that the binding is strong enough, it is necessary to know the outcome of the reaction. Instead of the complicated and tedious calculation of the ligand reaction result, it is much simpler to use neural networks. For ligand resolution, a dataset was constructed from the KEGG, RCSB PDB, PubChem and ZINC databases. The dataset contains a set of proteins and their substrates, inhibitors and activators. Each protein and ligand were transformed into an N-Cube representation and then added together with its protein to the neural network model.

The model uses convolutional learning, kernels (matrices used as filters in CNNs) are applied to both protein and ligand and then combined into a single linear layer with a uniform result.

The initial dataset was split into training, validation, and test datasets, with the final results summarized below. The result was surprising, achieving up to 97% accuracy (Fig. 5).

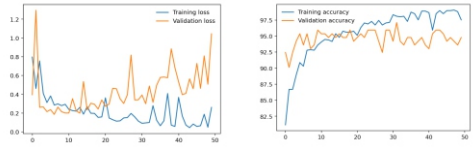


Fig. 5: Loss and accuracy of classifier training after transfer learning of convolutional networks

4. Molecular Docking to Detect Competitive Inhibition

During enzymatic reactions, both the binding of ligands to enzymes and the eventual conversion of these substrates into products and termination of binding occur. Using the equilibrium constant Keq we are able to represent both reactions. Here, I have chosen free energy for the calculation, as we would require complex simulations for the kinetic constants.

There are 3 basic methods to estimate the free energy itself. Using the full scheme requires an evaluation of the free energy at each point, and hence complex integrations. Using a finite point, like the full scheme, requires knowledge of the neighborhood of the ligand when it is outside the active region of the protein.

$$\Delta G_e = \frac{1}{2}\left[U_{site}^{elec} - U_{bulk}^{elec}\right] + \alpha\left[U_{site}^{vdW} - U_{bulk}^{vdW}\right] + \gamma$$

The best solution, however, is an empirical calculation that only considers the position of the ligand in the active region. The use of this calculation is mostly applied in molecular docking, which tries to determine the position of the ligand in the active site using this very rough approximation of the free energy. Docking scores can be calculated based on the Lennard- Jones potential (taking into account the interaction of atoms, mainly Van der Wals forces), electrostatic forces and solvation parameters (representing the displacement of water during the ligand reaction).

$$\Delta G_e = \sum_i^{prot}\sum_j^{lig}\left(\frac{A_{ij}}{r_{ij}^{12}} + \frac{B_{ij}}{r_{ij}^{6}} + 332\frac{q_i q_j}{\varepsilon(r_{ij})r_{ij}} + (V_i S_j + V_j S_i)e^{\frac{-r_{ij}}{2\delta^2}}\right)$$

$$A_{ij} = 4\varepsilon\delta^{12} \qquad B_{ij} = 4\varepsilon\delta^{6} \qquad S_i = L_i + 0{,}01097q_i \qquad V_i = \frac{4}{3}\pi r_i^3$$

$$\varepsilon(r_{ij}) = F + \frac{H}{\epsilon + ke^{-\mu H r_{ij}}}$$

5. NDock – Neural Molecular Docking

The dataset was compiled using the KEGG database. The protein structure (in both pdb and mmcif format), and subsequently the active site structure, was obtained from the RCSB PDB (Protein data bank) and the ligand structure from PubChem (in sdf format). The larger the dataset, the better the results can be expected. Unfortunately, I did not

have enough computing power to take full advantage of this fact, so I constructed a dataset of average size with the largest possible variation (approximately 3000 entries).

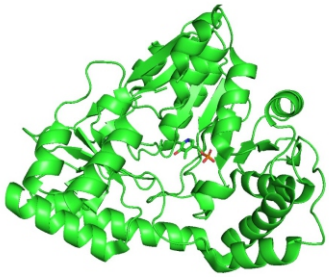


Fig. 6: Example of Protein and its ligand used from my dataset, PyMOL server was used for illustration

Before the data can be used to train a neural network, it must be normalized for a uniform representation. Thus, active sites and their ligands and targets were extracted from each protein. The target here is meant to be the position of the ligand given in the pdb file, the ligand is the position of the downloaded pdf file, so the goal is to get the ligand to the position of the target (Fig. 7).

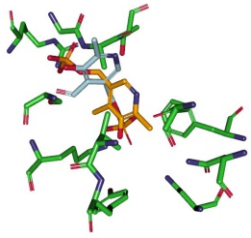


Fig. 7: Active site, target in orange, ligand in light blue

The ligand, target and active site are then normalized to the origin and rotated. The rotation is done by selecting the atom closest to the origin and then rotating it so that all dimensions, except y, are 0. Both the ligand and the active site are rotated in this way. The target is normalized and rotated by the same values as the active site.

The active site and ligand were converted to their respective representations (AMK and SELFIES sheets). A recurrent network was used for each part, which was pre-trained on the encoder/decoder model due to lack of computational resources. The encoder outputs were then combined into a single linear layer. The use of recurrent networks has the advantage that we can have an unspecified number of inputs.

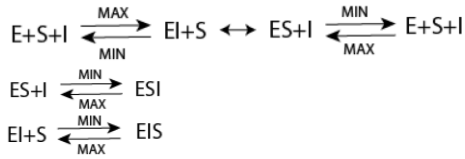
$$RMSD = \sqrt{\frac{1}{N_A}\sum_{i=0}^{N_A}((x_{Ri}-x_i)^2+(y_{Ri}-y_i)^2+(z_{Ri}-z_i)^2)}$$

The output of the model is 6 numbers, these represent the displacement of the centre of the molecule in space and rotation in 3 angles using rotation matrices. The loss of the model is then calculated using RMSD (Root mean square deviation).

6. Using Classification as a Reward System

The program uses feedback learning to modify the ligand and create a competitive inhibitor. However, in order to use feedback learning, we need a reward system by which the model can orient itself and learn how to proceed.

Initially, the ligand must be classified, differentiated into substrate, inhibitor or activator, indicating to the model that the ligand must be changed into an inhibitor. In the case of classification as an inhibitor, the score deals with the measurement of enzymatic reactions and tells us how good the inhibitor is competitively.



These reactions are evaluated based on a docking score (NDock used here) modelling the free energy and then combined in a final score, in case of a classification other than inhibitor, the program is terminated.

7. Results of the Molecule Modification Program

The goal of the program is rapid and accurate drug design. Therefore, three drugs that act as competitive inhibitors were selected, namely Acetazolamide, Viagra and Methotrexate (Fig. 8).

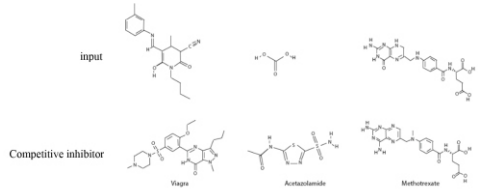


Fig. 8: Demonstration of input molecules, substrates, and experimentally determined competitive inhibitors

First, a dataset of their proteins, active sites and the substrates they inhibit was obtained. This data was entered into a program and 100 possible competitive inhibitors were obtained for each. The results were then divided into 2 categories, those that most closely resembled the experimental results and those that received the highest reward (Fig. 9).

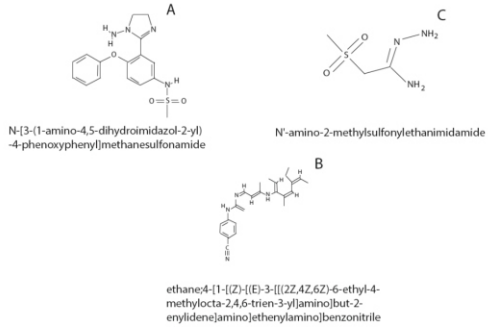


Fig. 9: Molecules with the highest score determined by the program. (A) Viagra. (B) Methotrexate. (C) Acetazolamide.

As can be seen here, the final results with the highest scores do not resemble their experimental counterparts. Even if they have higher programmed scores than the inhibitors themselves, it is difficult to say whether they are better, this could only be determined experimentally or by quantum chemical methods (Fig.10).

The molecules most similar to the experimental ones did not have the highest scores. These molecules, although bearing some similarity, started to form at the beginning of the program, then the further the program went on, the more

the molecules diverged. This phenomenon can also be explained by the fact that as the program goes on it gets better scores, so these molecules, even the experimental ones, are not the best. It's just that the properties of the ligand can be altered by a single changed atom, so concluding this assumption is rather wrong .

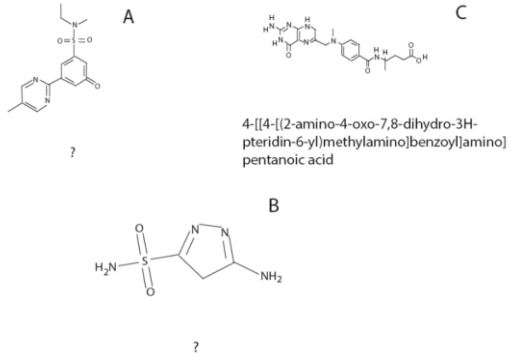


Fig. 10: Molecules with the highest similarity to the experimental results determined by the program. (A) Viagra. (B) Acetazolamide. (C) Methotrexate. Question marks indicate that the molecule was not found in the databases

The molecules generated by the program as possible competitive inhibitors of the previously mentioned substrates were finally compared with the database in PubChem and Zinc to see if they were valid and had already been synthesized. Of the said molecules, I just found 82% (2473 out of 3000 results) and was able to identify them. However, the remaining 18% passed as valid in a force field test to determine spatial arrangement .

8. Comparison of Neural Docking and Classical Methods

The carbanic anhydrase protein (in pdb format under 1dmy) in complex with the competitive inhibitor Acetazolamide, used for example in the treatment of epilepsy or glaucoma, was selected as an example of the results (Fig.11).

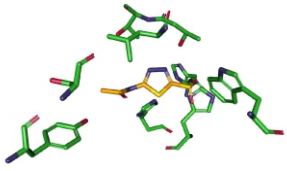


Fig.11: Example of docking target, crystallographically determined structure

To validate the classical methods, AutoDockTools software was first applied to add hydrogens, calculate charges, determine rotating ligand bonds, and determine the search space. The prepared protein and ligand were loaded into Autodock 4. The best result offered had an RMSD loss of 1.456 (Fig. 12).

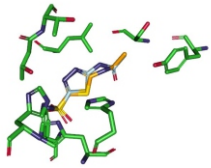


Fig.12: The result of molecular docking using Autodock 4 software, shown here in light blue

Subsequently, I repeated the process in my program. Here the final loss was 2.018 (Fig. 13).

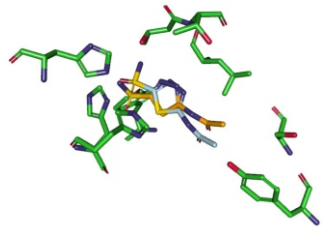


Fig. 13: Result of Neural Docking, here marked in light blue

Then I performed the same process on the extended dataset, each element of this data was not included in the training set for neural docking.

Table 1: Summary results for Autodock 4 and Ndock

Results	Autodock 4	NDock
RMSD	~2.045	~2,854
Time	~3 days	~12,13 min

Even though my loss was higher than the classical methods, it was still acceptable (the difference between the result and the target is such that the free energy calculations and other values are comparable to each other). The speed and efficiency, as expected, exceeded those of the classical methods. In the case of my program, the dataset of 1000 molecules and their proteins took me around a few minutes to run, Autodock 4 took several days.

9. Conclusion

The main goal of the work was to develop a program using deep feedback learning for the design of new competitive inhibitors. To achieve this goal, the program NDock was created using neural docking as a faster but not more accurate alternative to existing methods such as Autodock 4. In my work, I additionally created several libraries, mostly in the Python programming language.

Although NDock does not show the same accuracy as classical methods, its results are sufficient. However, its main advantage is its speed and low computational power cost. The program creates realistic, mostly already synthesized molecules. The library itself and its various functions, such as force fields, searching for non-covalent interactions or working with chemical files (sdf, pdb, mmcif), is applicable to other problems, but at the time of writing the library is far from complete.

The structure of the program itself is very flexible. If the score to be followed is defined correctly, the meaning and the goal can be easily changed. The work itself focuses only on competitive inhibitors, but the exact same procedure is applicable to other inhibitors, even other types of ligands. The idea that the input is the substrate, and the output is the inhibitor is also changeable. The input can be the experimental inhibitor itself and the function of the program becomes the modification of the inhibitor. In this way, the system can be adapted to any work with modification or creation of molecules.

References

[1] Shin, S.H., Bode, A.M. & Dong, Z. (2017). Precision medicine: the foundation of future cancer therapeutics. Dostupné z: https://doi.org/10.1038/s41698-017-0016-z

[2] Bayat A. (2002). Science, medicine, and the future: Bioinformatics. Available at: https://doi.org/10.1136/bmj.324.7344.1018

[3] Hughes, J. P., Rees, S., Kalindjian, S. B., & Philpott, K. L. (2011). Principles of early drug discovery. Dostupné z: https://doi.org/10.1111/j.1476-5381.2010.01127.x

[4] Kanehisa, M. and Goto, S. (2000). KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res. 28, 27-30.

[5] Kanehisa, M. (2019). Toward understanding the origin and evolution of cellular organisms. Protein Sci. 28, 1947-1951.

[6] Kanehisa, M., Furumichi, M., Sato, Y., Ishiguro-Watanabe, M., and Tanabe, M. (2021). KEGG: integrating viruses and cellular organisms. Nucleic Acids Res. 49, D545-D551.

[7] Kim S, Chen J, Cheng T, et al. (2021). New data content and improved web interfaces. Nucleic Acids Res. 2021;49(D1):D1388-D1395. Available at: https://doi:10.1093/nar/gkaa971

[8] Sterling, Irwin. (2015). J. Chem. Inf. Model. Available at: http://pubs.acs.org/doi/abs/10.1021/acs.jcim.5b00559

[9] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne. (2000). The Protein Data Bank. Available at: Nucleic Acids Research, 28: 235-242.

[10] RDKIT: Open-source cheminformatics. Available at: http://www.rdkit.org

[11] The PyMOL Molecular Graphics System, Version 1.2r3pre, Schrödinger, LLC.

[12] Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S. and Olson, A. J. (2009). Autodock 4 and AutodockTools4: automated docking with selective receptor flexibility. J. Computational Chemistry 2009, 16: 2785-91.

[13] Oystein Bjorke, Lunci Hua. (2012). Helix Toolkit WPF.

[14] Bruce Alberts, Alexander Johnson, Julian Lewis, David Morgan, Martin Raff, Keith Roberts a Peter Walter. Molecular biology of the cell. 6. ed. Canada, 2015, s. 1342. ISBN: 978-0-8153-4464-3.

[15] Johnson, K. A., & Goody, R. S. (2011). The Original Michaelis Constant: Translation of the 1913 Michaelis–Menten Paper. Biochemistry, 50(39), 8264–8269. Available at: https://doi.org/10.1021/bi201284

[16] Berg J., Tymoczko J. a Stryer L. (2002) Biochemistry. W. H. Freeman and Company. ISBN: 0-7167-4955-6.

[17] Ouertani A, Neifar M, Ouertani R, et al. Effectiveness of enzyme inhibitors in biomedicine and pharmacotherapy. Adv Tissue Eng Regen Med Open Access. 2019;5(3):85-90. Available at: https://medcraveonline.com/ATROA/effectiveness-of-enzyme-inhibitors-in- biomedicine-and-pharmacotherapy.html

[18] Diesenroth, Faisal a Ong. Mathematics for machine learning. 1. ed. Cambridge, 2020, s. 371. ISBN: 978-1-108-45514-5.

[19] Aurélian Géron-O'Reilly. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow. 2. ed. Canada, 2019, s. 819. ISBN: 9781492032649.

[20] Gilbert Strang. Introduction to linear algebra fifth edition. 5. ed. US, 2016, s. 573. ISBN: 978-0-9802327-7-6.

[21] Nie, F., Hu, Z., & Li, X. (2018). An investigation for loss functions widely used in machine learning. Commun. Inf. Syst., 18, 37-52. Available at: file:///C:/Users/-/Downloads/CIS18Aninvestigationforlossfunctionswidelyusedinmachinelearning.pdf

[22] James Stewart. Calculus: Early transcendentals. 7. ed. Brooks, 2008, s. 1170. ISBN: 978-0-538-49790-9.

[23] Jordan, M I. Serial order: a parallel distributed processing approach. Technical report, June 1985-March 1986. US: N. p., 1986. Web. Available at: https://cseweb.ucsd.edu/~gary/258/jordan-tr.pdf

[24] Sepp Hochreiter, Jurgen Schmidhuber (1997). Long short-term memory. Neural Computation. 9 (8): 1735-1780. S2CID 1915014. PMID 9377276. Available at: https://www.bioinf.jku.at/publications/older/2604.pdf

[25] Cho, Kyunghyun; van Marrinboer, Bart; Gulcehre, Caglar; Bahdanau, Dzmitry; Bougares, Fethi; Schwenk, Holder; Bengio, Yoshua (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Available at: https://arxiv.org/pdf/1406.1078.pdf

[26] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L.D. Jackel. Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4):541-551, 1989. Available at: http://yann.lecun.com/exdb/publis/pdf/lecun-89e.pdf

[27] Sun, Z., Ozay, M., & Okatani, T. (2015). Design of Kernels in Convolutional Neural Networks for Image Classification. Available at: http://yann.lecun.com/exdb/publis/pdf/lecun- 89e.pdf

[28] Mouton, Coenraad; Myburgh, Johannes C.; Davel, Marelie J. (2020). Gerber, Aurona (ed.). Stride and Translation Invariance in CNNs. Artificial Intelligence Research. Communications in Computer and Information Science. Cham: Springer International Publishing. 1342: 267-281. Available at: https://arxiv.org/pdf/2103.10097.pdf

[29] Gholamalinejad, H., & Khosravi, H. (2020). Pooling Methods in Deep Neural Networks, a Review. Available at: http://yann.lecun.com/exdb/publis/pdf/lecun-89e.pdf

[30] Boué, L. (2018). Deep learning for pedestrians: backpropagation in CNNs. Available at: https://arxiv.org/ftp/arxiv/papers/1811/1811.11987.pdf

[31] Francois, V., Henderson, P., Islam, R., Bellemare, M., & Pineau, J. (2018). An Introduction to Deep Reinforcement Learning. Foundations and Trends® in Machine Learning, 11, 219–354. doi:10.1561/2200000071. Available at: https://arxiv.org/pdf/1811.12560.pdf

[32] Richard S. Sutton a Andrew G. Barto. Reinforcement Learning: An Introduction. 2. ed. Cambridge, 2015, s. 352. Available at: https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf

[33] Burnetas, A., & Katehakis, M. (1997). Optimal Adaptive Policies for Markov Decision Processes. Mathematics of Operations Research - MOR, 22, 222–255. Available at: file:///C:/Users/-/Downloads/MDPS_mor97.pdf