Reminders to finish last exercises and to our way of working: -1, 0
Prepare before class such that you can present your result: 3, 4, 5
Worked on in class, but you can prepare them at home: 1, 2, 6

Carefully read the instructions below!

(-1) Finalize last exercise sheet

Finalize all exercises of the last exercise sheet and upload the results to your defined GitHub repository.

(0) Setup a git project on GitHub

In order to get a bit of `git` training done we work for all the exercises on GitHub.

(a) Use the project created for the previous exercises or create a new **private** project in GitHub

(b) Give the instructor (kandolfp) access to the project (see GitHub documentation for help)

(c) Create a `pdm` project in your repository and commit the necessary files.

(d) Create an appropriate structure in your repository for the rest of the exercise sheet (maybe have a look at the exercises to have a better idea first), not everything should be in the main folder.

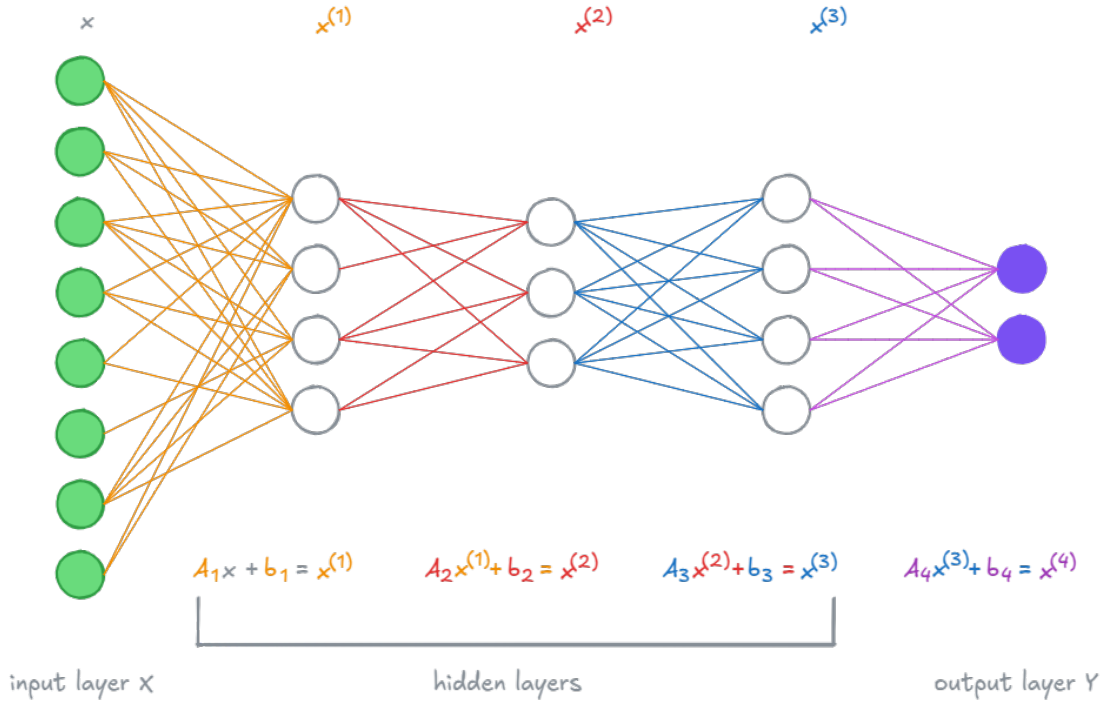
(e) Try to structure your work on the exercises with git, i.e.

- Don't commit things that do not belong together in one single commit. Each exercise can be considered as a separate thing. Subparts of an exercise might be independent as well.
- Use meaningful commit messages `https://www.conventionalcommits.org/en/v1.0.0/`
- Make sure that you do not commit something that does not work - produces an error. If you have difficulties with an exercise you can also commit your best effort in this case.

(f) Add a README.md that explains what you are doing, how to run the exercises and anything else that is necessary (quick guide to `pdm`), maybe note your name somewhere.

(g) Optional: Work with issues, you can reference the issue in the commit message, GitHub documentation

(1) Compute the parameters of the Neural Network



For the generic NN display above answer the following questions:

(a) What are the shapes of the matrices A_1, A_2, A_3, A_4?

(b) Which of these matrices are sparse (have multiple zeros)?

(c) What are the shapes of the biases b_1, b_2, b_3, b_4?

(d) Write down the (expanded) formula for the output y with respect to the input x resulting from the composition of the different layers for $f(x) = x$ in each layer (as seen in the above image).

(e) Is this formulation a good option to compute y from x (provide some reasoning)?

(2) Sizes and parameters of a model

For the sequential model created in Section 6.3.2 of the lecture notes, use the

```
model.summary(line_length=80,
              expand_nested=True, show_trainable=True)
```

To produce a text based summary of the model and answer the following questions:

(a) Explain how the `Param #` column is computed and retrace the details.

(b) What does the memory usage seen in brackets after *Total params:* and *Trainable params* imply on the used number format for the parameters?

(c) What does the `(None, 10)` imply for the shape?

(3) Optimizers for the training

As we have seen so often before, the optimizer used for the problem can change how fast convergence is reached (if at all).

(a) Have a look at the possibilities provided in the framework Overview - Optimizers Keras documentation - they are basically all improvements on Stochastic Gradient Descent.

(b) Test different approaches, especially one of the Adam (`Adam`, `AdamW`, `Adamax`) and Ada (`Adadelta`, `Adafactor`, `Adagrad`) implementations and record the performance (final accuracy).

(4) Model persistence

Time end resources run into training a model. Therefore we need to make sure a model can be persisted aka. written on the disc. How can you save and load a keras model? Try with some model from the lecture or these exercises and see if the inference is any different after loading the model again.

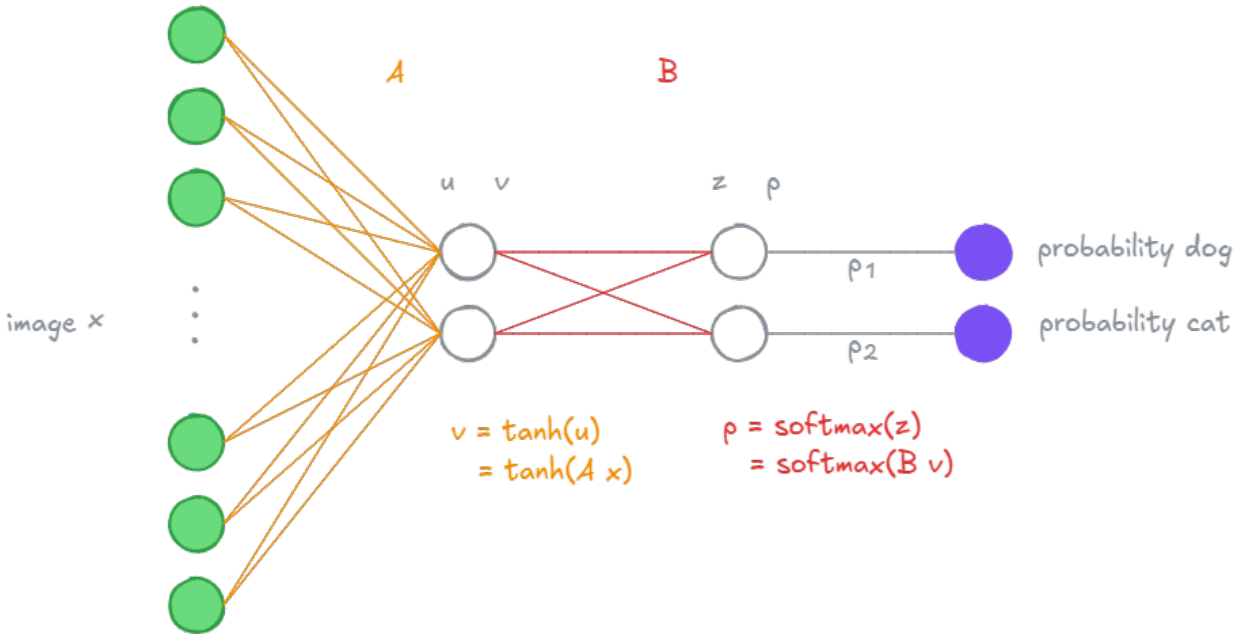
(5) Data augmentation for images

Quite often it is advisable to (randomly) augment the data for training and sometimes also for inference. It has shown to be an important technique to improve several aspects of neural networks, like robustness, overfitting, generalization, and more. You can find the methods include in keras via image augmentation in the documentation.

For any of the discussed CNNs for images from the lecture, apply appropriate transformation to the training dataset and see how this influences the performance for the testset. As a reference, see Infobox in Section 7.6.2 form the lecture notes.

(6) Backpropagation with a simple example

To illustrate the procedure we start with the simplest example



In order to do so, we need to specify our variables in more detail. To simplify it a bit we set the biases to the zero vector.

First, we call our output $p = [p_1, p_2]$ and our labels are encoded in one-hot encoding $y = [y_1, y_2]$ where $y_1 = 1$ and $y_2 = 0$ if the image is a dog, and vice versa if the image is a cat.

As our loss function we use cross-entropy

$$\begin{aligned}\mathscr{L}(\Theta) &= -\frac{1}{2}\sum_{i=1}^{2} y_i \log(p_i) = -\frac{y_1 \log(p_1) + y_2 \log(p_2)}{2} \\ &= -\frac{1}{2}\left(y_1 \log(p_1) + (1 - y_1)\log(1 - p_1)\right),\end{aligned}$$

for a single sample with the above notation. The last line is true to the fact that the sum of the entries of y and p is equal to 1.

I n order to make the computation of the derivatives easier we use the variables as described in the above image.

Therefore, we get $p = g(z) = \sigma(Bv)$ (softmax), and $v = f(u) = \tanh(Ax)$. Overall we want to compute the change in $B_{i,j}$ (for some fixed indices i, and j).

$$\frac{\partial \mathscr{L}}{\partial B_{i,j}} = \frac{\partial \mathscr{L}}{\partial p} \cdot \frac{\partial p}{\partial z} \cdot \frac{\partial z}{\partial B_{i,j}}$$

Perform this task in the following steps:

(a) Compute $\partial_{p_i} \mathscr{L}$.

(b) The computation of the Jacobian of $\sigma(z)$ is tricky but together with the cross-entropy loss it is straight forward, therefore compute $\partial_{z_i} \mathscr{L}(\sigma(z))$.

(c) Compute $\partial_{B_{i,j}} z$.

(d) Write down the components showing up for the chain rule for $\frac{\partial \mathscr{L}}{\partial A_{i,j}}$ (similar as above).

Try to solve these problems on your own. If you get stuck, maybe some of your fellow students can help you. Why not write down your question in the course forum in SAKAI. Please try not to write back entire solutions or exchange solutions in the forum, this is not the idea and will backfire in the long run (also we might decide to delete such posts).