(https://profile.intra.42.fr)

Remember that the quality of the defenses, hence the quality of the of the school on the labor market depends on you. The remote defences during the Covid crisis allows more flexibility so you can progress into your curriculum, but also brings more risks of cheat, injustice, laziness, that will harm everyone's skills development. We do count on your maturity and wisdom during these remote defenses for the bene fits of the entire community.

SCALE FOR PROJECT MULTILAYER-PERCEPTRON (/PROJECTS/42CURSUS-MULTILAYER-PERCEPTRON)

You should evaluate 1 student in this team

Git repository

Introduction

In order to maintain high evaluation standards, you are expected to:

- Remain polite, courteous, respectful and constructive at every moment of the discussion. Trust between you and our community depends on your behaviour.

- Highlight the flaws and issues you uncover in the turned-in work to the evaluated student or team, and take the time to discuss every aspect extensively.

- Please take into account that discrepancies regarding the expected work or functionalities definitions might occur. Keep an open mind towards the opposite party (is he or she right or wrong?), and grade as honestly as possible. 42's pedagogy only makes sense if peer-evaluations are carried out seriously.

Guidelines

- You must grade only what exists in the GiT repository of the student or team.

- Remember to check the GiT repository's ownership: is it the student's or team's repository, and for the right project?

- Check thoroughly that no wicked aliases have been used to trick you into grading something other than the genuine repository.

- Any script supposed to ease the evaluation provided by one party must be thoroughly checked be the other party in order to avoid unpleasant situations.

- If the student in charge of the grading hasn't done the project yet, it is mandatory that he or she reads it before starting the evaluation.

- Use the available flags on this scale to tag an empty work, a non functional work, a coding style ("norm") error if applicable, cheating, and so on. If a flag is set, the grade is 0 (or -42 in case of cheating). However, cheating case excluded, you are encouraged to carry on discussing what went wrong, why, and how to address it, even if the grading itself is over.

Attachments

 evaluation.py (/uploads/document/document/2113/evaluation.py)

 subject.pdf (https://cdn.intra.42.fr/pdf/pdf/13338/en.subject.pdf)

 data.csv (/uploads/document/document/2114/data.csv)

Implementation

This section is the evaluation of the program implementation.

Modularity

Does the artificial neural network contains at least 2 hidden layers ? (which results in a network with a depth of at least 4 counting the input and ouput layers)

 Yes No

Dataset split

Is the dataset split in two, with a part for the training and another for the validation ? (The validation set is used to

determine the performances of the model on unknown examples).

Yes No

Softmax function implementation

Is the softmax function implemented on the neural network's ouput layer ? This is important as the softmax function returns the output as a probabilistic distribution.

Yes No

Training

This section is the evaluation of the training phase. Execute the program and grade the following questions.

Display

Is something going on ?

- Metrics are displayed at the end of each epoch ? (similar to the example in the subject)
- The training curves are displayed at the end of the training ? (there must be at least 2 curves, the training and the validation curve)

Yes No

Metrics

Is there at least one metric for the evaluation on the training set and one for the validation set ? Are the metrics adapted for binary classification ? (if it's the mean square error, root mean square error, binary cross-entropy loss for example, it's good, otherwise ask for further explanations).

Yes No

Model

How is the training doing ?

- Does the model learn ? (the value of the training and validation metrics get better in general the more the model trains)
- The model is not overfitting.

 Yes No

Saving

Is the model saved at the end of the training ? (Both the network topology and the weights must be saved)

 Yes No

Prediction

This section is the evaluation of the prediction phase of the model on the test set. Execute the program and grade the following questions.

Loading

The prediction program must load the weights and topology of the network as well as the dataset on which the predictions will be performed.

- The trained model is correctly loaded (check that the values are not hard-coded for example). - The test set is correctly loaded.

 Yes No

Metrics

The prediction program evaluates the performance of the model on the test set with the binary cross-entropy loss function (if it's not the case, the next question can not be evaluated).

Yes No

Model performances

Perform the following instructions :

- Download the program evaluation.py from the project resources. - Execute the program with `python evaluation.py` (this program downloads the dataset, shuffles it and splits it in two). - Train a new model with the dataset `data_training.csv`. - Perform a prediction on the dataset `data_test.csv` with this model.

You can train up to 3 models in order to do the evaluation of this question and keep only the best prediction (training a model depends on a lot of random factors such as the weights initialization, as such different trainings converge to different solutions).

Look at the best prediction obtained on the test set,

- The value is above 0.35 => 0
- The value is between 0.35 and 0.25 => 1
- The value is between 0.25 and 0.18 => 2
- The value is between 0.18 and 0.13 => 3
- The value is between 0.13 and 0.08 => 4
- The value is below 0.08 => 5

Rate it from 0 (failed) through 5 (excellent)

Algorithms understanding

You are going to evaluate the student's understanding of the concepts and algorithms at the heart of artificial neural networks. Try to be objective during this evaluation and give all the points if you feel that the student really grasps the subject.

Feedforward

Ask the student to explain the feedforward algorithm.

Rate it from 0 (failed) through 5 (excellent)

Gradient descent

Ask the student to explain the gradient descent algorithm.

Rate it from 0 (failed) through 5 (excellent)

Backpropagation

Ask the student to explain the backpropagation algorithm.

Rate it from 0 (failed) through 5 (excellent)

Overfitting

Ask the student to explain what is a situation of overfitting.

Rate it from 0 (failed) through 5 (excellent)

Bonus

The bonus section must be evaluated only if the mandatory part is perfectly done.

Bonus

Attribute one point per bonus. A bonus corresponds to a minimum of time investment from the student, functionalities that take 5 minutes to add don't deserve points.

Rate it from 0 (failed) through 5 (excellent)

Ratings

Don’t forget to check the flag corresponding to the defense

Ok

Outstanding project

Empty work

Incomplete work

Invalid compilation

Cheat

Crash

Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation