Apprentissage Statistique Automatique II

Lucas Reding, Thibault Defourneau

Lecture note

1 Regularization

Exercises 1. *Train a classifier on a dummy dataset containing 2000 data points and 2 classes, using the method* `make_classification from sklearn.datasets`.

1. *Build the classifier with one dense layer of 10 units with a ReLU activation function and a final softmax layer, using Stochastic Gradient Descend with learning rate* 0.01 *and binary cross entropy loss.*
2. *Train this model on the dummy dataset using train and validation data points with* 100 *epochs.*
3. *Plots the evolution of loss on train and validation data*
4. *Re-train the model but using l_2 regularization on the intermediate dense layer.*
5. *What difference do you observe with and without l_2 regularization ?*
6. *Make the same experiment with a dropout and with batch normalization.*

Exercises 2. *Train a neural network on the dataset CIFAR formed by 60,000 images with 10 classes, which can be loaded via the method $tf.keras.datasets.cifar10.load_data$:*

1. *Build a neural network with* 20 *layers where each layer contains* 100 *neurons and Swish as activation function.*
2. *Train the model on the dataset with $Nadam$ optimizer and early stopping. Experiment different learning rates.*
3. *Add batch normalization between the layers, and train the new model. Does the model converge faster ? Is the trained model better ?*

2 Convolutional Neural Networks

Exercises 3. *We now study the CNN called VGG-11.*

1. *Build this neural network using Tensorflow.*
2. *What is the total number of parameters to train ?*

Exercises 4. *Build your own CNN from scratch using Tensorflow, and try to find the most acurate one on MNIST dataset.*

Exercises 5. *Use transfer learning for large image classification, going through these steps:*

1. *Create a training set containing at least 100 images per class from the dataset $tf_flowers$, which can be obtained from* `tensorflow_datasets`. *For example, you could classify your own pictures based on the location (beach, mountain, city, etc.), or alternatively you can use an existing dataset (e.g., from TensorFlow Datasets).*
2. *Split it into a training set, a validation set, and a test set.*
3. *Build the input pipeline, including the appropriate preprocessing operations, and optionally add data augmentation.*
4. *Fine-tune a pretrained model on this dataset.*

3 Large Language Models

For the following exercises, we are going to use the library `transformers`, which is very popular.

Exercises 6. *Generate an answer based on the prompt "Create a funny joke about chickens." using the model* `microsoft/Phi-3-mini-4k-instruct`*. Propose two manners to perform the same task.*

Exercises 7. *Build a recommendation system using only dense embeddings based on the dataset located at*

https://www.kaggle.com/datasets/harshshinde8/movies-csv

Given a prompt, it should recommend the top 5 most similar movies. What strategies can be done for improving the recommendation system ?

Exercises 8. *Based on the same dataset as in the previous exercise, can you group the movies based only on the columns 'keywords' and 'overview' ?*